IDS 572

Assignment 2

Case– Loan default prediction and investment strategies in online lending

Ketaki Rane

Sanidhya Armarkar

Raj Mehta

Part A

1. (a) Your team's ultimate goal is to help clients determine whether they should invest in p2p loans. What is the final decision that you will help the client make? What is the objective, and how will you evaluate 'better' vs 'worse' decisions? What is the goal of predictive models for this ? What will be the potential target variables?

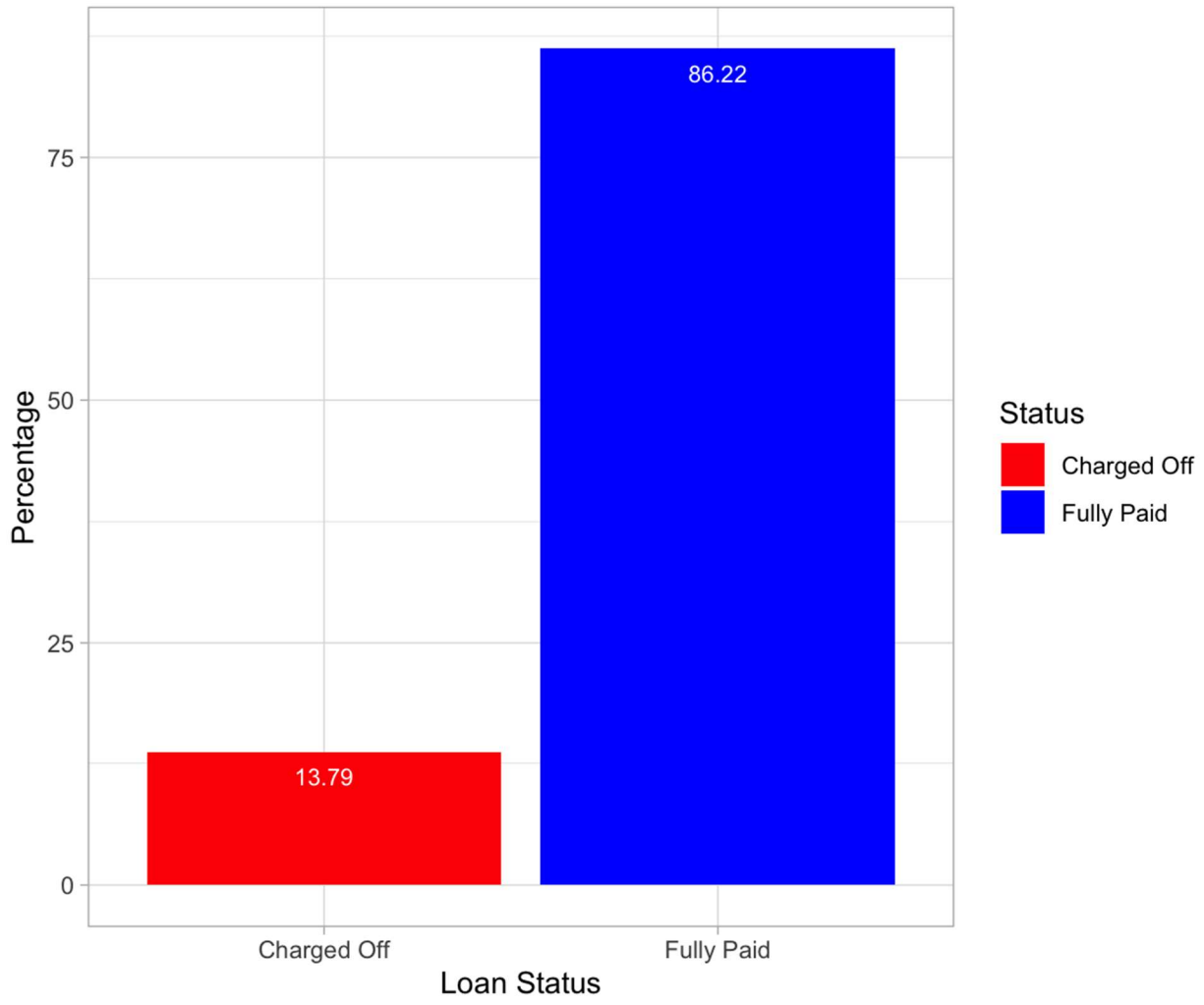
We will help clients to decide if they should invest in a particular loan or not by analyzing key connections between data. These relationships include the loan's objective and any connections to performance it could have, as well as a review of the returns on the loans.
We can decide which scenario is "better" or "worse" by identifying a gain or loss that is anticipated to occur upon loan payback. We will calculate the average gain or loss of fully paid loans and compare it with each loan to determine whether investing is profitable.
Predictive models all have three main objectives: helping clients make decisions about their loan investments, helping clients understand the risk involved with P2P platforms, and helping clients realize their expected profits. Loan amount, loan grade, number of payments, loan status, and other factors could be the target variables.
By computing the average gain or loss of fully paid and charged-off loans, you may determine if investing in each loan is lucrative. If the profit margin for loans that have been fully repaid is greater than the profit margin for loans that have not been repaid, the consumer will be able to decide with certainty whether to invest in either type of loan.

(b) Take a look at the data attributes. How would you categorize these attributes, in broad terms, considering what they pertain to? Before doing any analyses, what do you think may be the important attributes to consider for your decision task?

The data attributes, in broad terms can be categorized as loan characteristics, borrower characteristics, platform outcome. Loan characteristics such as interest rate, installment, loan start, loan end, purpose, loan amount etc. are the attributes that explain loan terms and what it is used for. Borrower characteristics such as annual income, credit score, total payment etc.These characteristics are unquestionably highly beneficial because they enable investors to determine whether the borrowers are capable of reliably repaying their investment. Platform outcomes include loan grade, term, subgrade. A loan's performance is influenced by a number of factors, including loan status, frequency of payments, and others. The loan amount and interest rate are regarded as important variables because they must be taken into consideration while making a decision. The loan's performance will be assessed based on its current status and payment history.

2. Data exploration (a) some questions to consider:
(i) What is the proportion of defaults ('charged off' vs 'fully paid' loans) in the data? How does the default rate vary with loan grade? Does it vary with sub-grade? And is this what you would expect, and why?

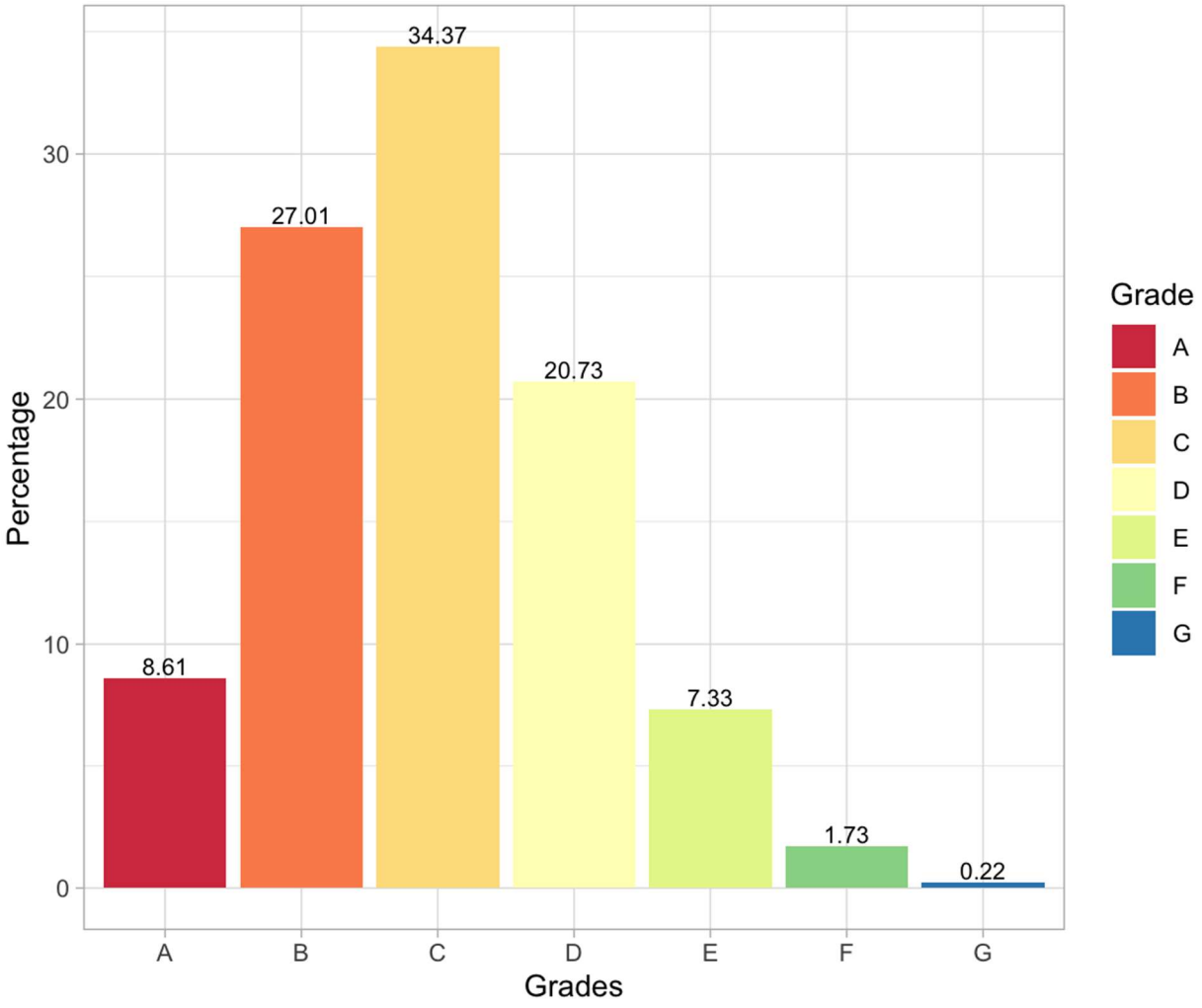
- What is the proportion of defaults ('charged off' vs 'fully paid' loans) in the data?



Loan status vs Percentage

- A total of 86.22% of the loans are fully repaid, while 13.79% have been charged off or are in default.

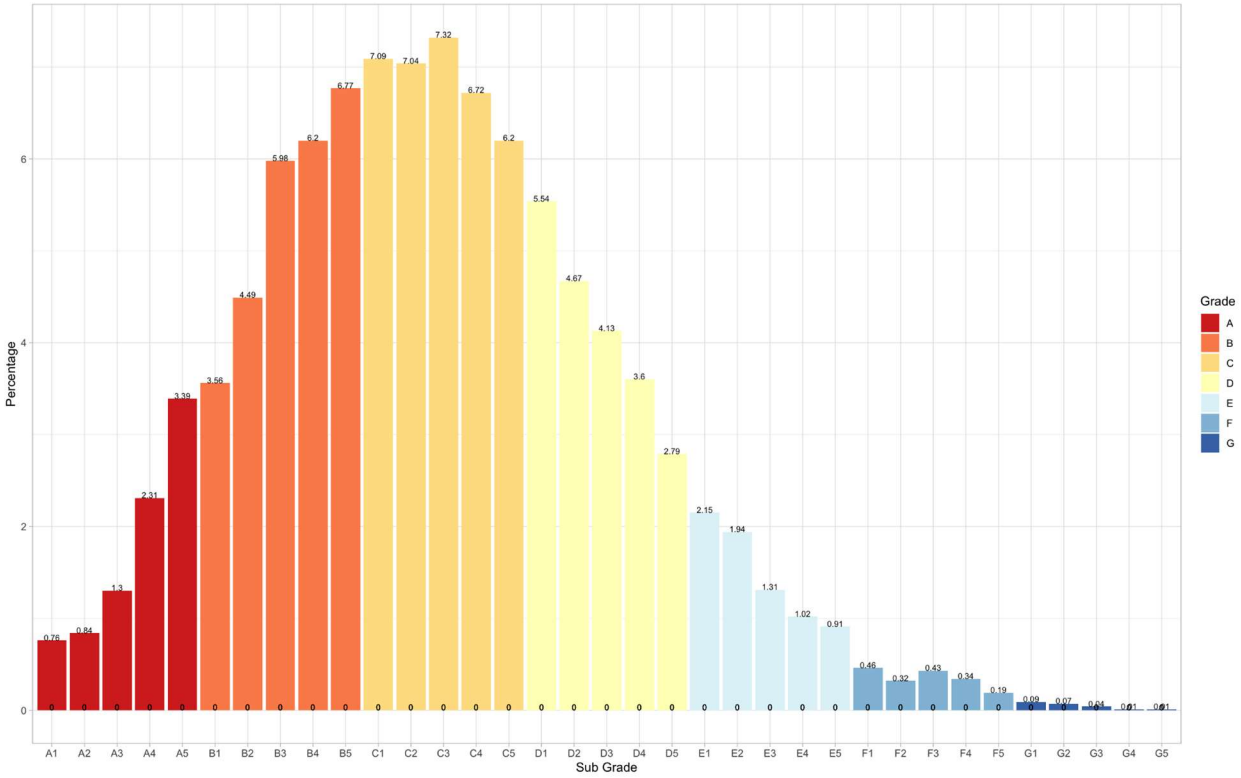
❖ How does the default rate vary with loan grade?



Grades vs Percentage

- The percentage of defaulters is highest in grade C at 34.37 percent, followed by grades B and D at 27.01 percent and 20.73 percent, respectively.
- Grade G has the lowest default rate of 0.22 percent.

❖ Does it vary with sub-grade?



SubGrade vs Percentage

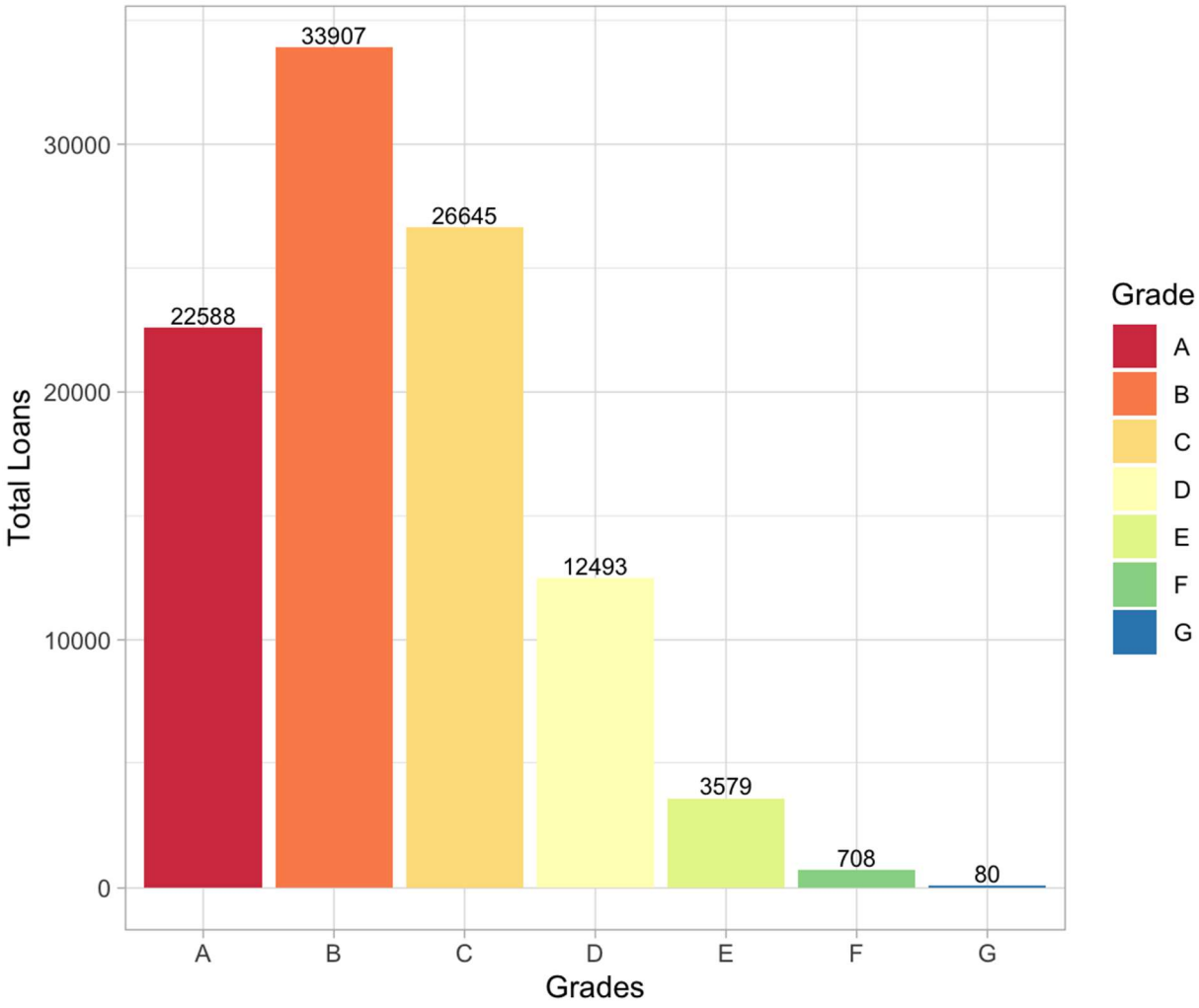
- The graph illustrates how default rates change with subgrades. As students move from grade A1 to A5 and grade B1 to B5, respectively, we observe a significant rise in the default rates for grades A and B. As students move from D1 to D5 and E1 to E5, the default rates for grades D and E decrease.
- Grades C and E do not, however, consistently improve or deteriorate as we move from C1 to C5 and E1 to E5, respectively.

❖ Is this what you would expect, and why?

- The intended outcome was not this. Despite expectations that grade G would have the most records, grades B, C, and D actually have more overall due to their distinct distributions.

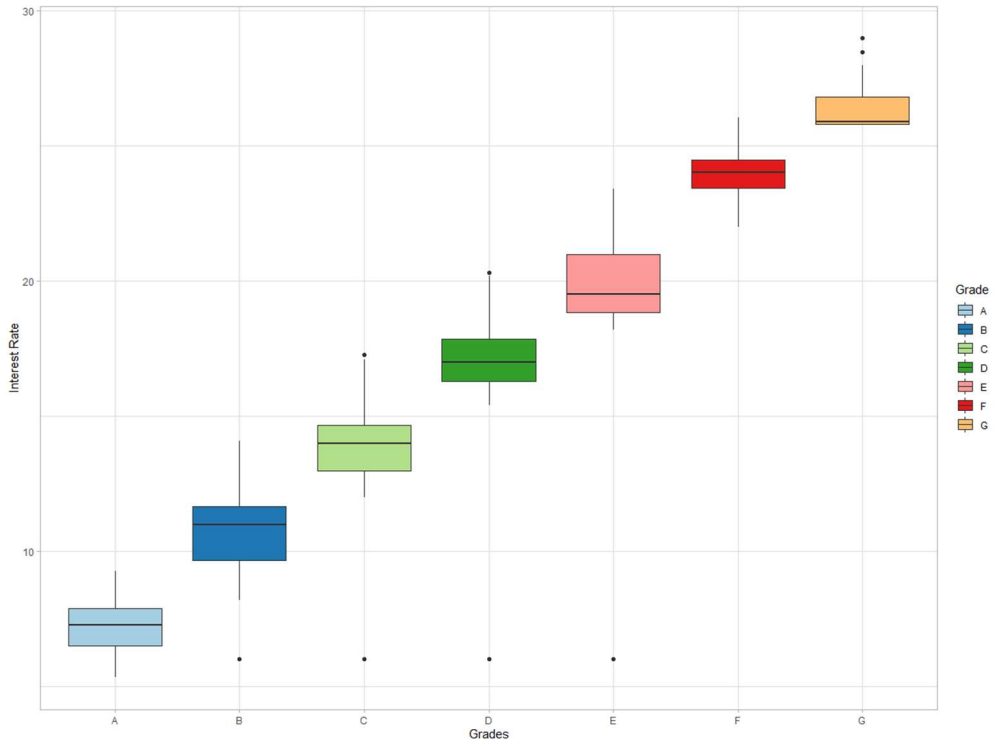
(ii) How many loans are there in each grade? And do loan amounts vary by grade? Does interest rate for loans vary with grade, subgrade? Look at the average, standard-deviation, min and max of interest rate by grade and subgrade. Is this what you expect, and why?

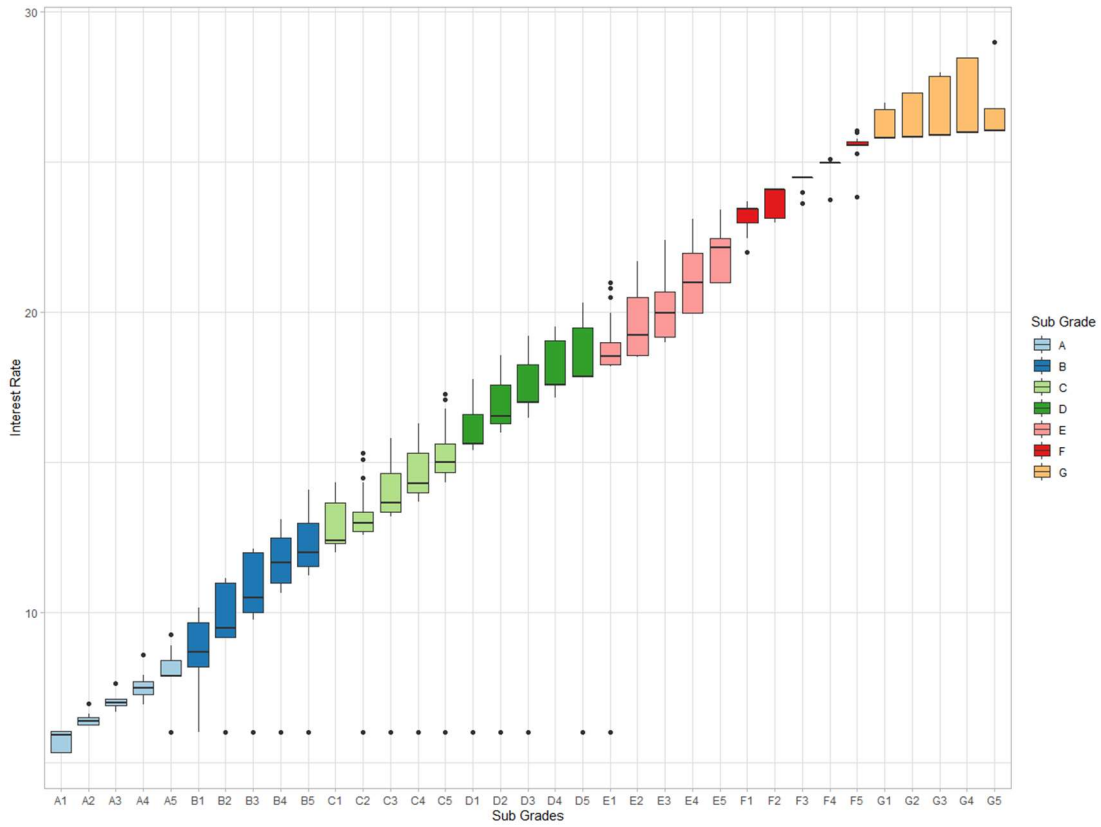
❖ How many loans are there in each grade?



- The graph above shows the total number of loans for each grade. According to grade, the quantity of loans varies; there is an upward trend from A to B and a downward trend from B to G.

❖ Does interest rate for loans vary with grade, subgrade?





According to the graphs, there is a noticeable correlation between interest rates and grades.

From point A to point G, the interest rates increase. Subgrades follow a similar pattern.

The average, standard deviation, minimum and maximum interest rates by grade and subgrade should be analyzed.

	grade	avgIR	sdIR	minIR	maxIR
1	A	7.17	0.967	5.32	9.25
2	B	10.8	1.44	6	14.1
3	C	13.8	1.19	6	17.3
4	D	17.2	1.22	6	20.3
5	E	19.9	1.38	6	23.4
6	F	24.0	0.916	22.0	26.1
7	G	26.4	0.849	25.8	29.0

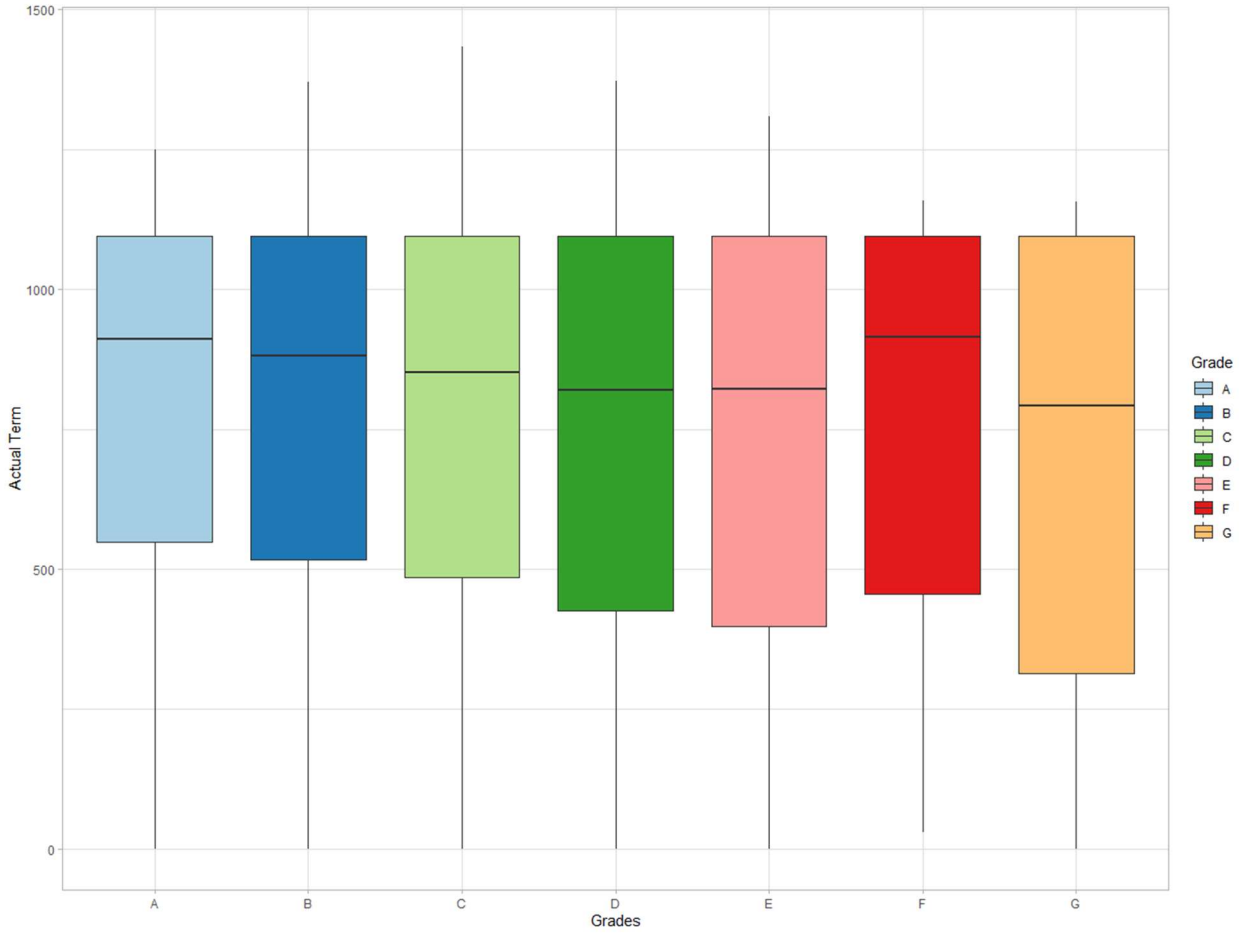
	sub_grade	avgIR	sdIR	minIR	maxIR
1	A1	5.68	0.347	5.32	6.03
2	A2	6.42	0.166	6.24	6.97
3	A3	7.09	0.325	6.68	7.62
4	A4	7.48	0.357	6.92	8.6
5	A5	8.24	0.424	6	9.25
6	B1	8.87	0.722	6	10.2
7	B2	9.96	0.816	6	11.1
8	B3	10.8	0.887	6	12.1
9	B4	11.7	0.840	6	13.1
10	B5	12.2	0.851	6	14.1

- ❖ Is this what you expect, and why?

• Yes, as the grades ranged from A to G and we thus believed that as the grade rose, so would the interest rate. Loan interest rates ought to be greater for borrowers with a G than for those with an A.

(iii) For loans which are fully paid back, how does the time-to-full-payoff vary? For this, calculate the 'actual term' (issue-date to last-payment-date) for all loans. How does this actual-term vary by loan grade (a box-plot can help visualize this).

- For loans which are fully paid back, how does the time-to-full-payoff vary?. How do these actual terms vary by loan grade?



As we progress from grade A to F, the length of time it takes to return the loan lowers. But for G it is higher.

(iv) Calculate the annual return. Show how you calculate the percentage annual return. Is there any return from loans which are 'charged off'? Explain. How does return from charged - off loans vary by loan grade? Compare the average return values with the average interest-rate on loans – do you notice any differences, and how do you explain this? How do returns vary by grade, and by sub-grade. If you wanted to invest in loans based on this data exploration, which loans would you invest in?

- Calculate the annual return. Show how you calculate the percentage annual return. Is there any return from loans which are 'charged off'?

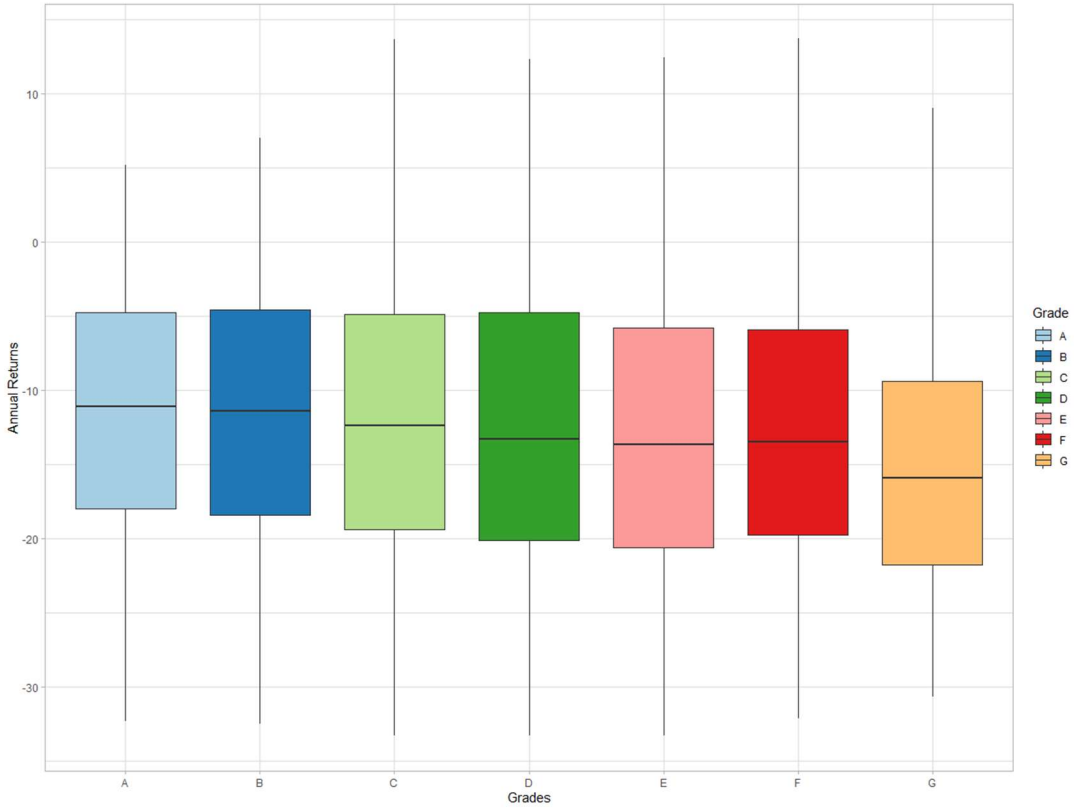
The annual total return percentage can be calculated by using the below formula.
annRet = ((total_pymnt - funded_amnt)/funded_amnt)*(12/36)*100

Yes, some loans offer an annual return that is positive.

The annual return varies for all loan classes, from A to G, depending on whether the loan is fully charged or charged off. Because the lending company lost money and only a small number of recoveries were made, the % yearly return for any clients who were charged off from any grade is negative for charged-off loans.

```
> lcdf %>% group_by(loan_status) %>%summarise(avgRec=mean(recoveries))
# A tibble: 2 x 2
  loan_status avgRec
  <chr>         <dbl>
1 Charged Off   926.
2 Fully Paid      0
> |
```

❖ How does return from charged - off loans vary by loan grade?



• The annual yield on a charged-off loan declined as we progressed from A to G. And it was to be expected.

- ❖ Compare the average return values with the average interest rate on loans – do you notice any differences, and how do you explain this?

	loan_status	intRate	totRet
1	Charged Off	13.9	-35.9
2	Fully Paid	11.7	15.4

• Because borrowers are not repaying the loan amount, it was expected that the total annual return values for the charged-off would be negative.

	grade	intRate	totRet
1	A	7.17	7.18
2	B	10.8	8.84
3	C	13.8	8.48
4	D	17.2	8.67
5	E	19.9	7.69
6	F	24.0	9.11
7	G	26.4	3.72

• The interest rate grows from grade A to G, but there is a variation in the growth for total return rates because the borrowers do not repay the amount.

	sub_grade	intRate	totRet
1	A1	5.68	6.50
2	A2	6.42	6.99
3	A3	7.09	7.33
4	A4	7.48	7.11
5	A5	8.24	7.66
6	B1	8.87	8.40
7	B2	9.96	8.84
8	B3	10.8	8.77
9	B4	11.7	9.38
10	B5	12.2	8.76

We can see that interest rates are rising in the sub-grades, as expected. And, as with grades, a similar pattern of deviation is observed in total return values.

Negative returns imply that the investment was lost because there are no returns from charged-off loans save for recoveries. When comparing charged off vs. completely paid loans,

the annual return percentage and average interest paid by clients show that Lending Club is losing nearly twice as much money, or the profits from fully paid customers are equal to the losses from charged off status customers.

- ❖ If you wanted to invest in loans based on this data exploration, which loans would you invest in?

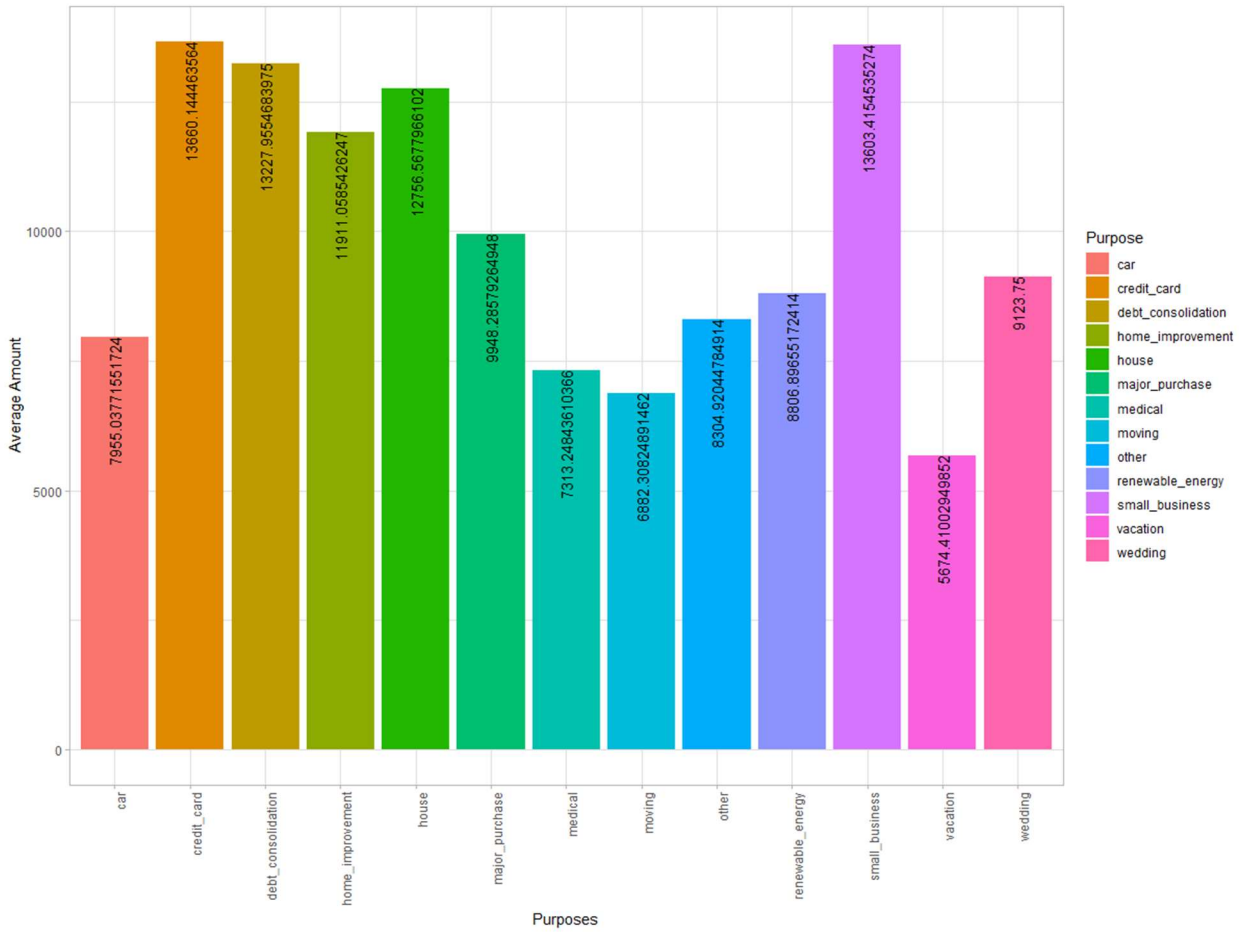
If we wanted to invest in loans based on data exploration, F1 and G3 (high risk) would be our best bet due to the high return values.

(v)What are people borrowing money for (purpose)? Examine how many loans, average amounts, etc. by purpose? Do loan amounts vary by purpose? Do defaults vary by purpose? Does loan-grade assigned by Lending Club vary by purpose?

```
> pur_tbl<-lcdf %>% group_by(purpose) %>% summarise(nLoans=n(),
+                                                   sumAmnt=sum(loan_amnt),avgAmnt=mean(loan_amnt),
+                                                   avgIntr= mean(int_rate),defaults=sum(loan_status=="Charged Off"),
+                                                   defaultRate=defaults/nLoans)
> pur_tbl
# A tibble: 13 x 7
   purpose             nLoans   sumAmnt avgAmnt avgIntr defaults defaultRate
   <chr>                <int>     <int>   <dbl>   <dbl>    <int>       <dbl>
 1 car                    928   7382275   7955.    11.5      107       0.115
 2 credit_card          24989 341353350  13660.    10.6     2865       0.115
 3 debt_consolidation   57622 762221250  13228.    12.2     8319       0.144
 4 home_improvement      5654  67345125  11911.    11.8      682       0.121
 5 house                  354   4515825  12757.    15.3       63       0.178
 6 major_purchase        1823  18135725   9948.    12.1      266       0.146
 7 medical               1119   8183525   7313.    14.3      172       0.154
 8 moving                 691   4755675   6882.    16.1      144       0.208
 9 other                 5091  42280350   8305.    14.7      838       0.165
10 renewable_energy        58    510800   8807.    15.7       11       0.190
11 small_business         893  12147850  13603.    16.8      203       0.227
12 vacation               678   3847250   5674.    14.5      101       0.149
13 wedding                100    912375   9124.    18.0       14       0.14
> |
```

• The table above outlines the various reasons that customers borrowed money. There are additional statistics on the quantity of loans, the total amount, the average interest rate, the total number of defaults, and the percentage of defaults for each purpose.
• The bulk of loan defaults are caused by debt consolidation loans, then credit card loans.
• Additionally, more people who were in a fully paid position than those who were charged off borrowed money for their debt consolidations. The graph below illustrates how the loan amount varies depending on the purpose and the loan status (charged off or fully paid). The highest and second-highest loan amounts in Fully Paid are for Credit Card and Debt Consolidation, respectively. The Charged Off follows a similar trend.

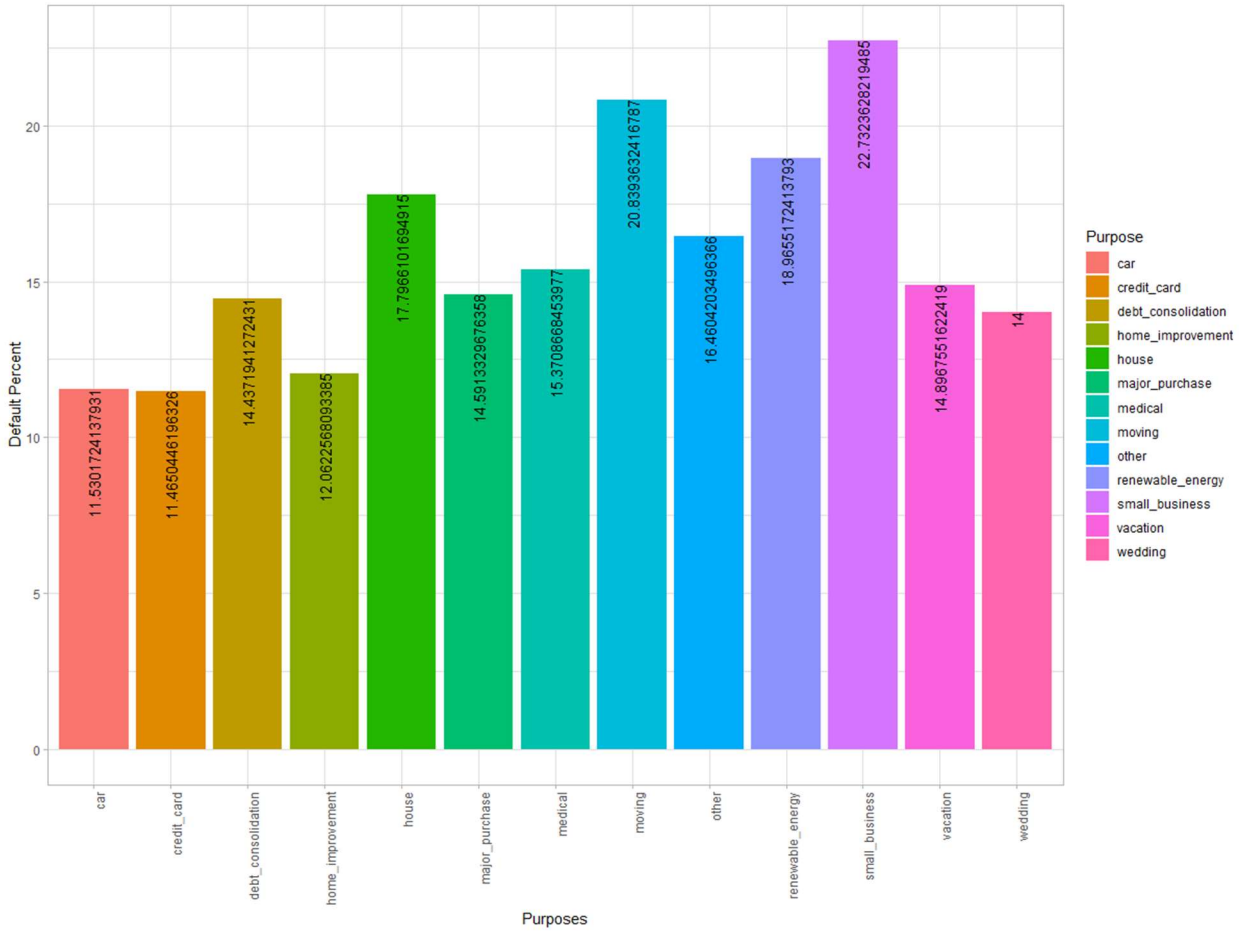
Do loan amounts vary by purpose?



•The graph above demonstrates how loan amounts vary based on the purpose. Credit cards and small businesses come out on top when the loan amounts are sorted by purpose, followed by debt consolidation.

❖ Do defaults vary by purpose?

```
> dflt_pur_tbl<-lcdf %>% group_by(purpose) %>% summarise("N_coff"=sum(loan_status=='Charged Off'),
+                                                        "N_fPaid"=sum(loan_status=='Fully Paid'),
+                                                        "prcnt_defalut"=(sum(loan_status=='Charged Off')/(sum(loan_status=='Charged Off') +
+                                                                                                            sum(loan_status=='Fully Paid'))*100))
> dflt_pur_tbl
# A tibble: 13 x 4
   purpose             N_coff N_fPaid prcnt_defalut
   <chr>                <int>   <int>         <dbl>
 1 car                    107     821          11.5
 2 credit_card           2865   22124          11.5
 3 debt_consolidation    8319   49303          14.4
 4 home_improvement       682    4972          12.1
 5 house                   63     291          17.8
 6 major_purchase         266    1557          14.6
 7 medical                172     947          15.4
 8 moving                 144     547          20.8
 9 other                  838    4253          16.5
10 renewable_energy        11      47          19.0
11 small_business         203     690          22.7
12 vacation               101     577          14.9
13 wedding                 14      86          14
> |
```



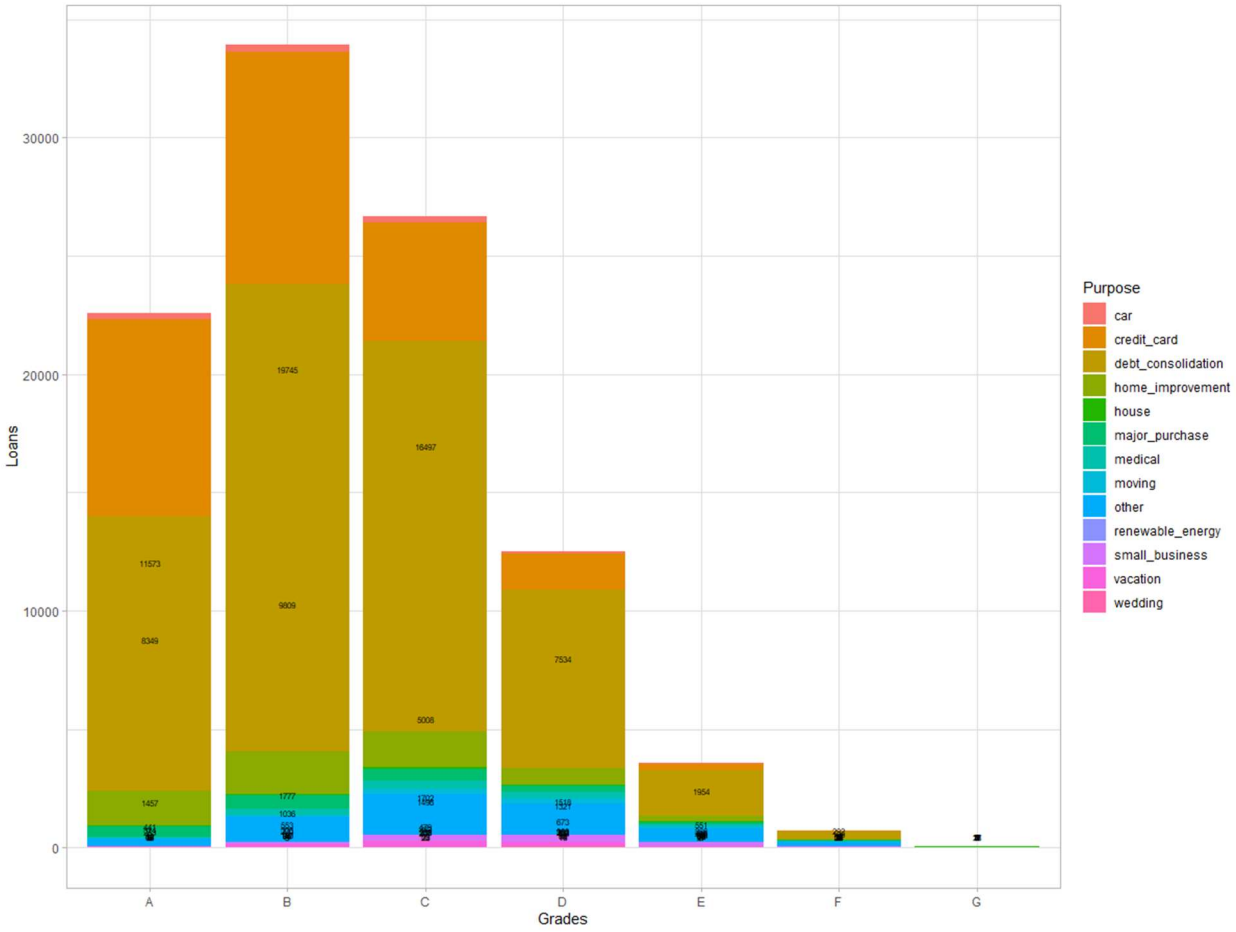
• Moving has an 18.14 percent share, renewable energy has 20 percent, and small businesses have 23.18 percent when we filter the defaults by purpose.

•The remaining percentages vary from 11.53 percent for cars to 17.26% for medical expenses.

❖ Does loan-grade assigned by Lending Club vary by purpose?

purpose	grade A	B	C	D	E	F	G
car	253	306	238	92	27	8	4
credit_card	8349	9809	5008	1518	266	37	2
debt_consolidation	11573	19745	16497	7534	1954	292	27
home_improvement	1457	1777	1496	673	215	33	3
house	37	74	83	74	48	27	11
major_purchase	441	553	479	252	70	26	2
medical	84	270	382	251	97	34	1
moving	10	96	207	234	108	32	4
other	324	1036	1702	1321	551	139	18
renewable_energy	3	5	22	18	8	2	0
small_business	15	100	249	300	159	62	8
vacation	42	127	257	180	59	13	0
wedding	0	9	25	46	17	3	0

>



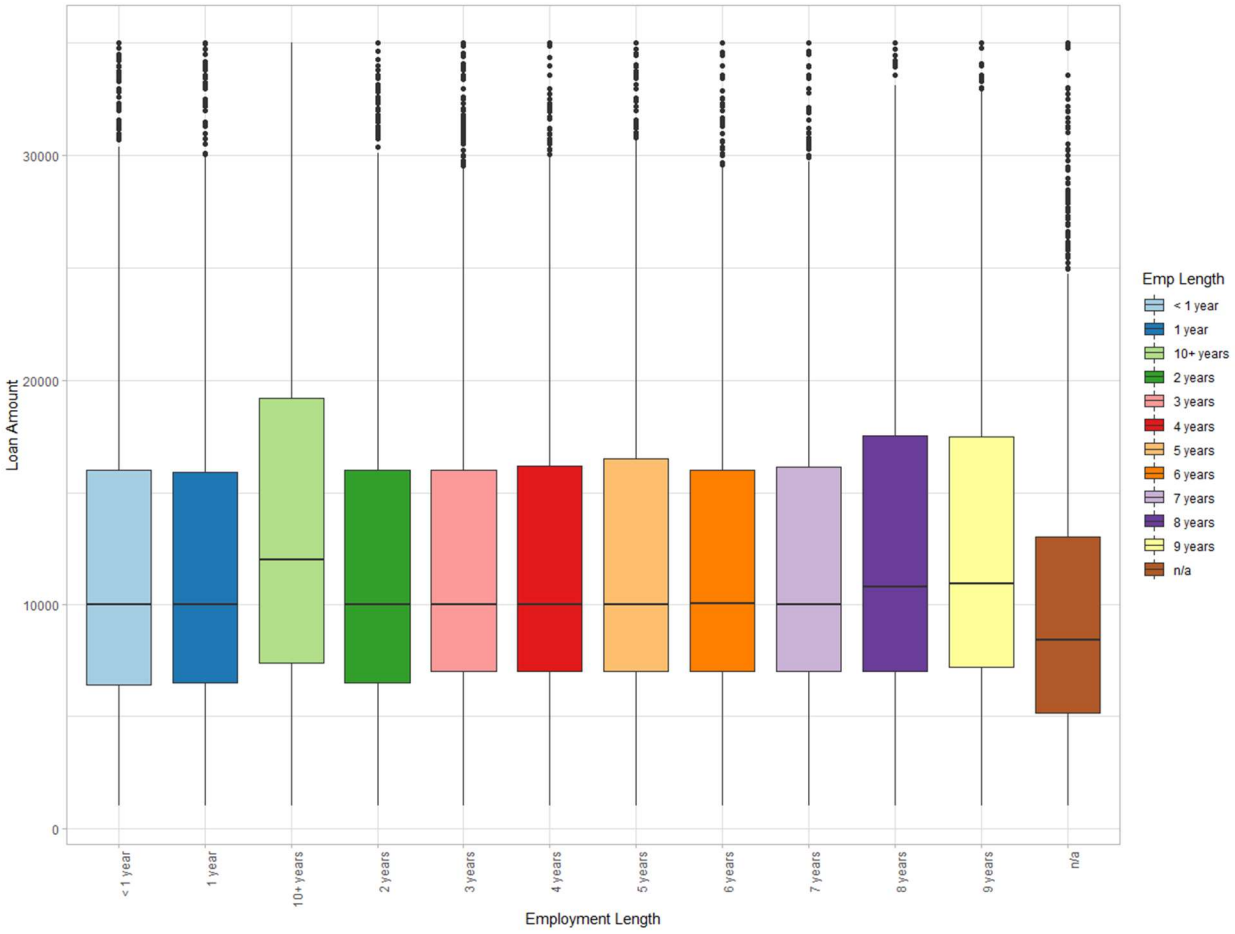
• To ensure that they were not biased, Lending Club gave each of the purposes a unique grade. In other words, the club is being fair when grading each purpose, and no one goal is given a benefit.

(vi) Consider some borrower characteristics like employment-length, annual-income, fico-scores (low, high). How do these relate to loan attributes like, for example, loan_amout, loan_status, grade, purpose, actual return, etc.

- ❖ In order to offer consumers credit or loans, certain borrower criteria are essential. Therefore, the status of those who pay more and owe less may rely on attributes like length of employment, annual income, and fico scores (before issuing a loan).

- ❖ Consider some borrower characteristics like employment-length, annual-income, fico-scores (low, high). (Low, High). How do these relate to loan attribute like, for example, loan_amount, loan_status, grade, purpose, actual return, etc.

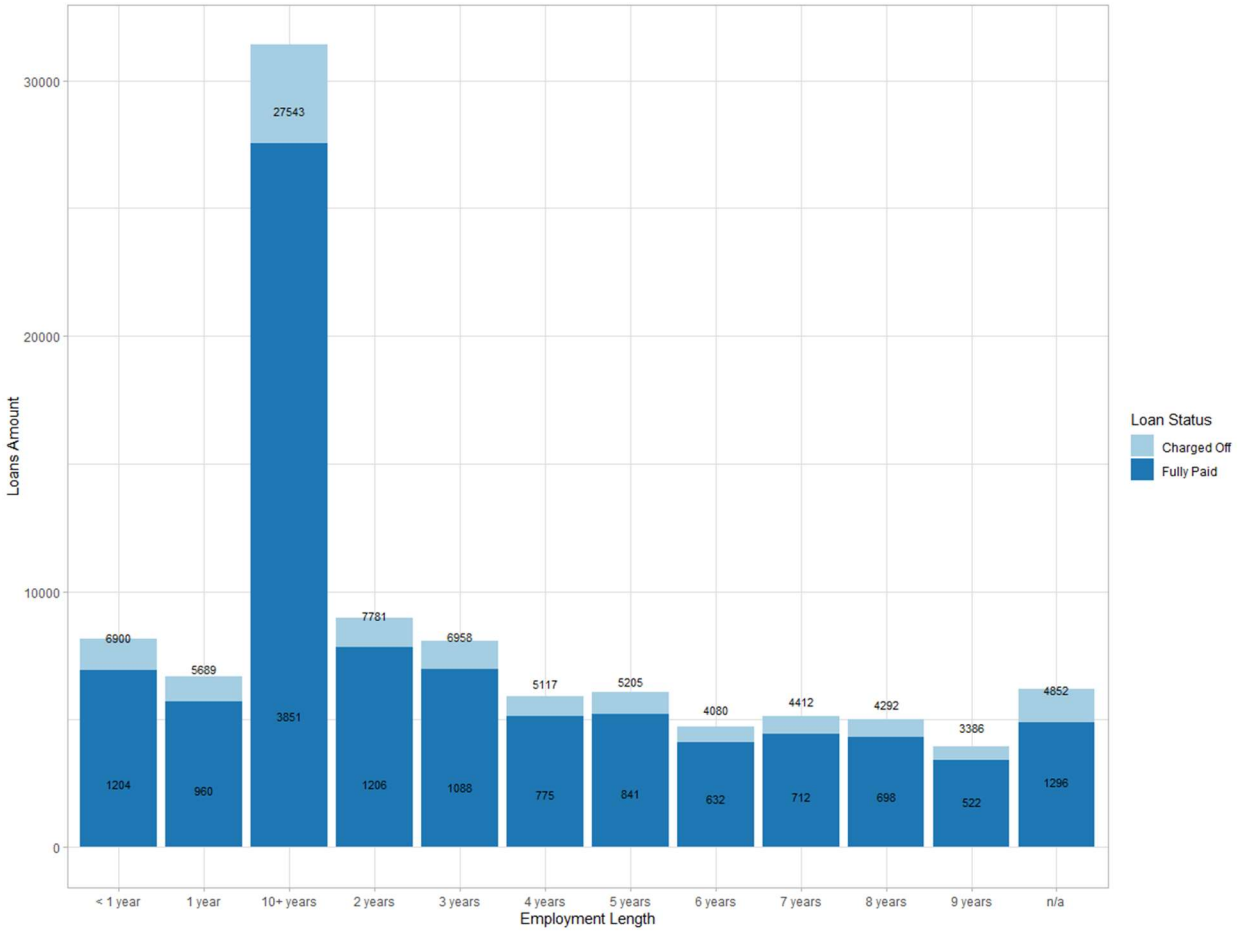
vii-a:employment-length vs loan_amount



•It is clear that the loan amount slightly rises as employment duration does (1-10+ years).

vii-b:employment-length vs loan_status

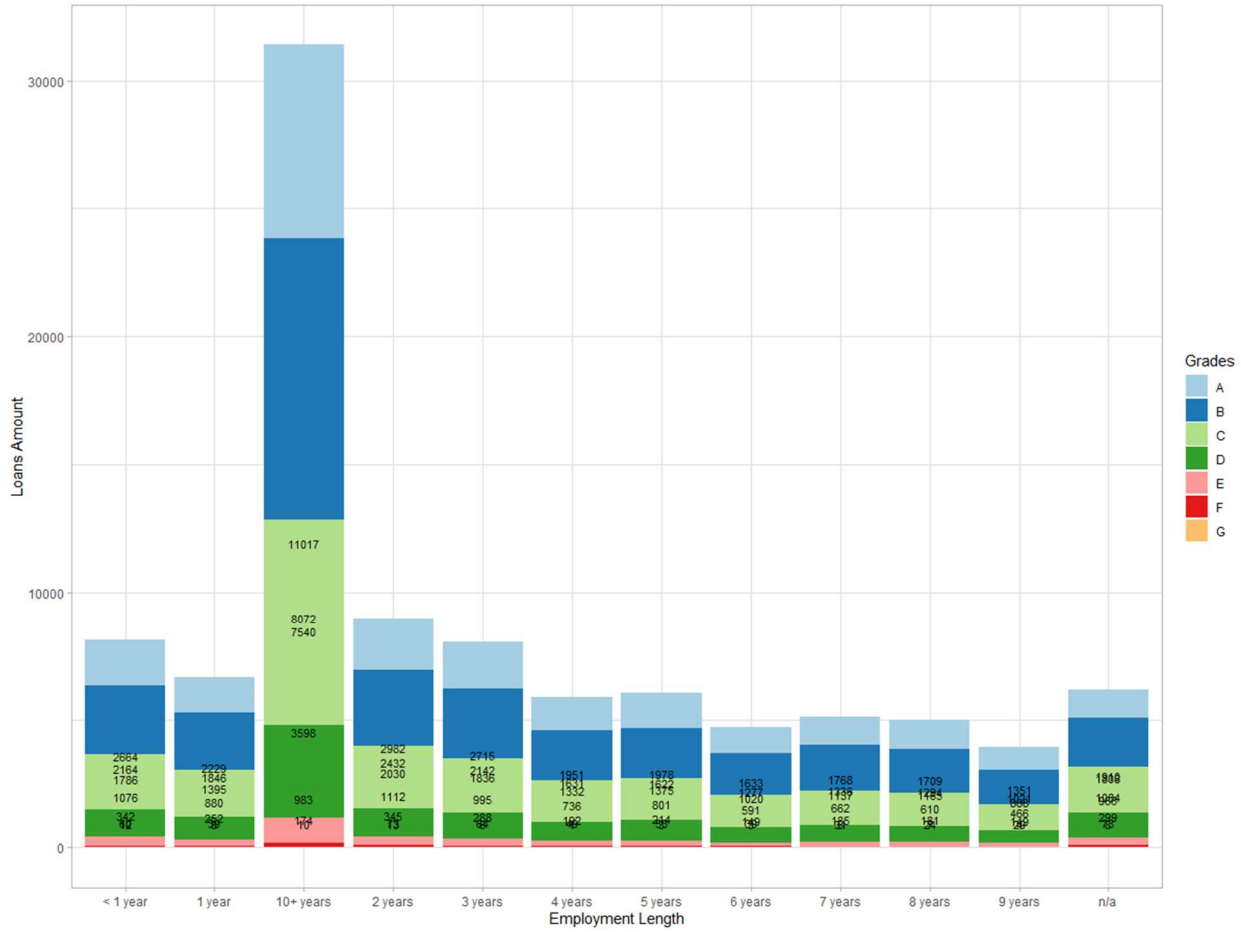
emp_length	loan_status Charged Off	Fully Paid
< 1 year	1204	6900
1 year	960	5689
10+ years	3851	27543
2 years	1206	7781
3 years	1088	6958
4 years	775	5117
5 years	841	5205
6 years	632	4080
7 years	712	4412
8 years	698	4292
9 years	522	3386
n/a	1296	4852



•Despite the fact that many applicants have 10 years or more of work experience, the default rate remains constant regardless of the duration of employment.

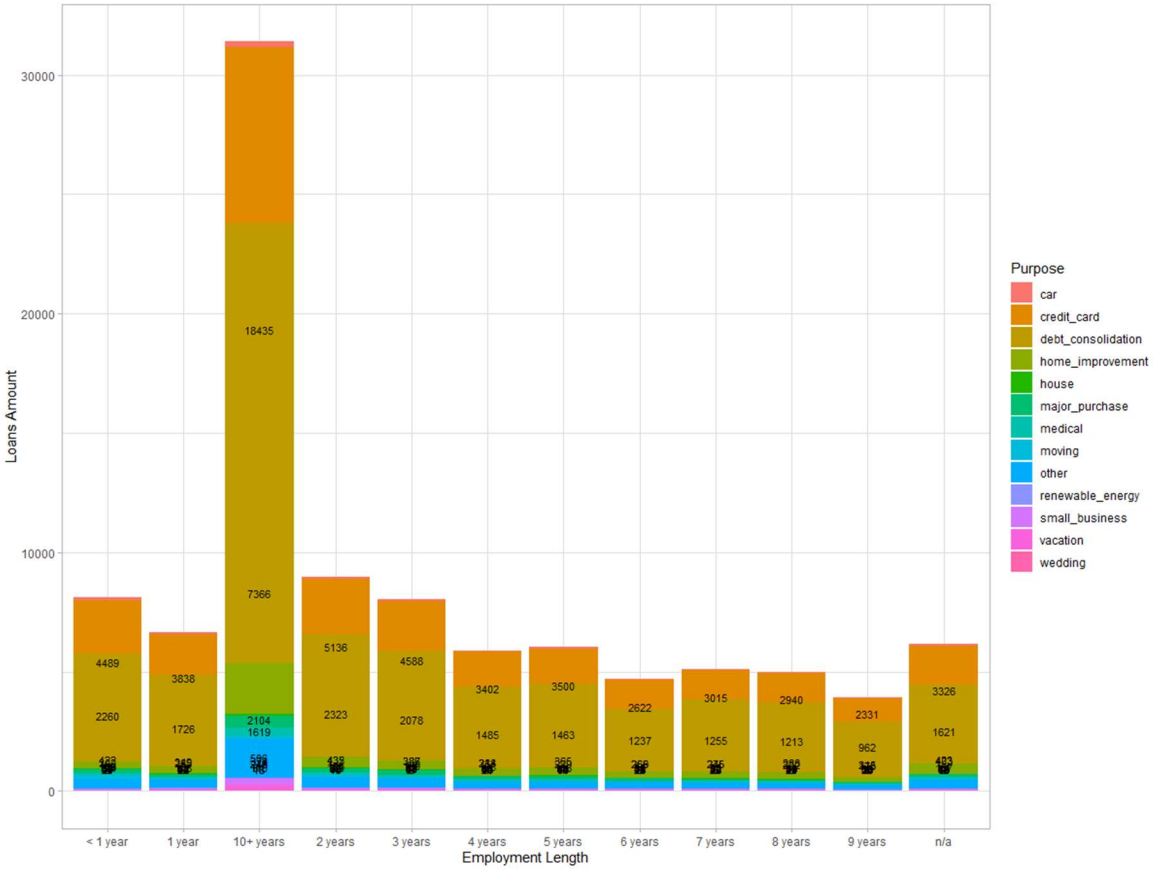
vii-c: employment-length vs grade

emp_length	grade A	B	C	D	E	F	G
< 1 year	1786	2664	2164	1076	342	60	12
1 year	1395	2229	1846	880	252	39	8
10+ years	7540	11017	8072	3598	983	174	10
2 years	2030	2982	2432	1112	345	73	13
3 years	1836	2715	2142	995	288	64	6
4 years	1332	1951	1631	736	192	49	1
5 years	1375	1978	1622	801	214	53	3
6 years	1020	1633	1277	591	149	39	3
7 years	1137	1768	1336	662	185	31	5
8 years	1165	1709	1294	610	181	24	7
9 years	888	1351	1021	466	149	29	4
n/a	1084	1910	1808	966	299	73	8



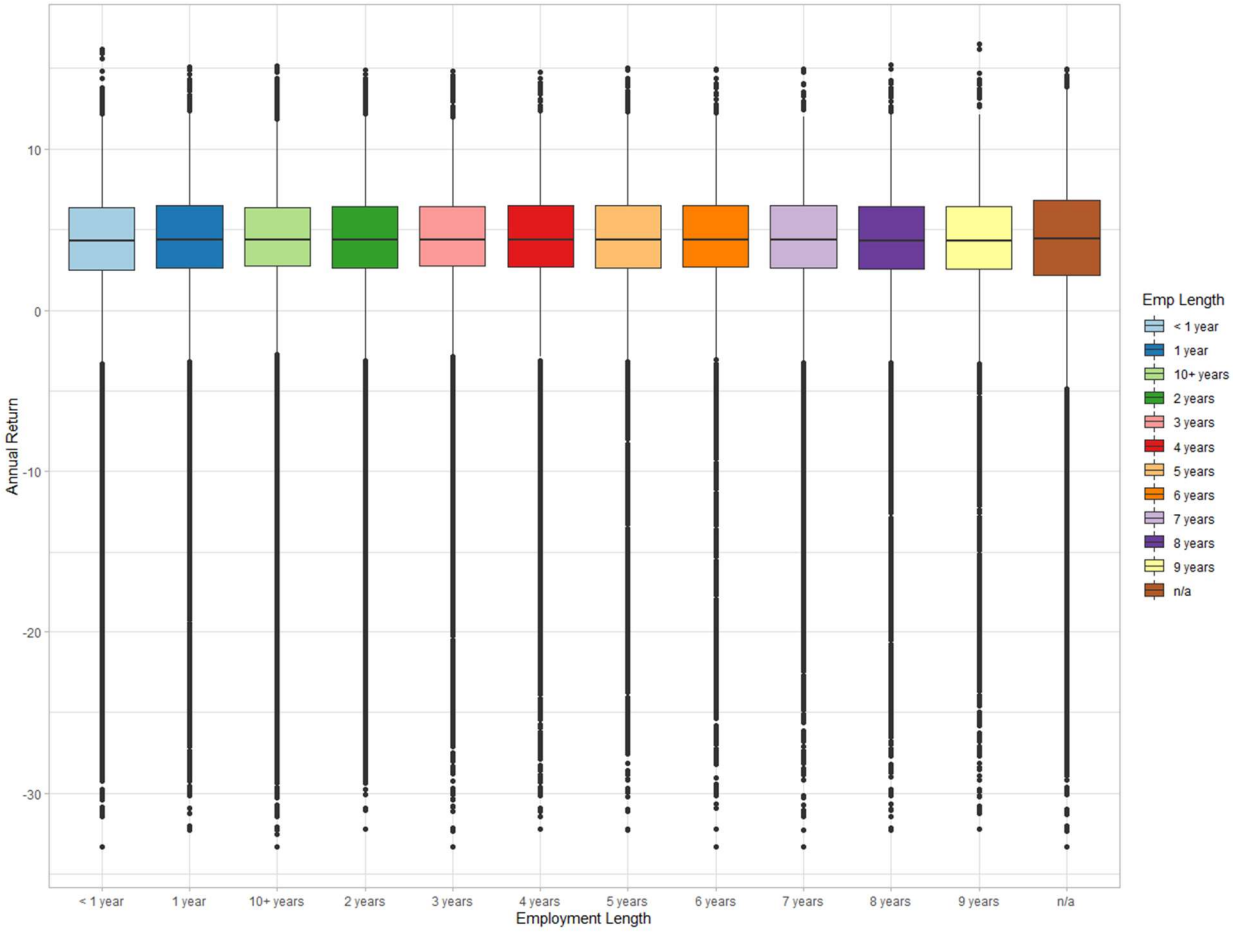
• Within each specific grade, the percentage stays the same over the course of all employment periods.

vii-d: employment-length vs purpose



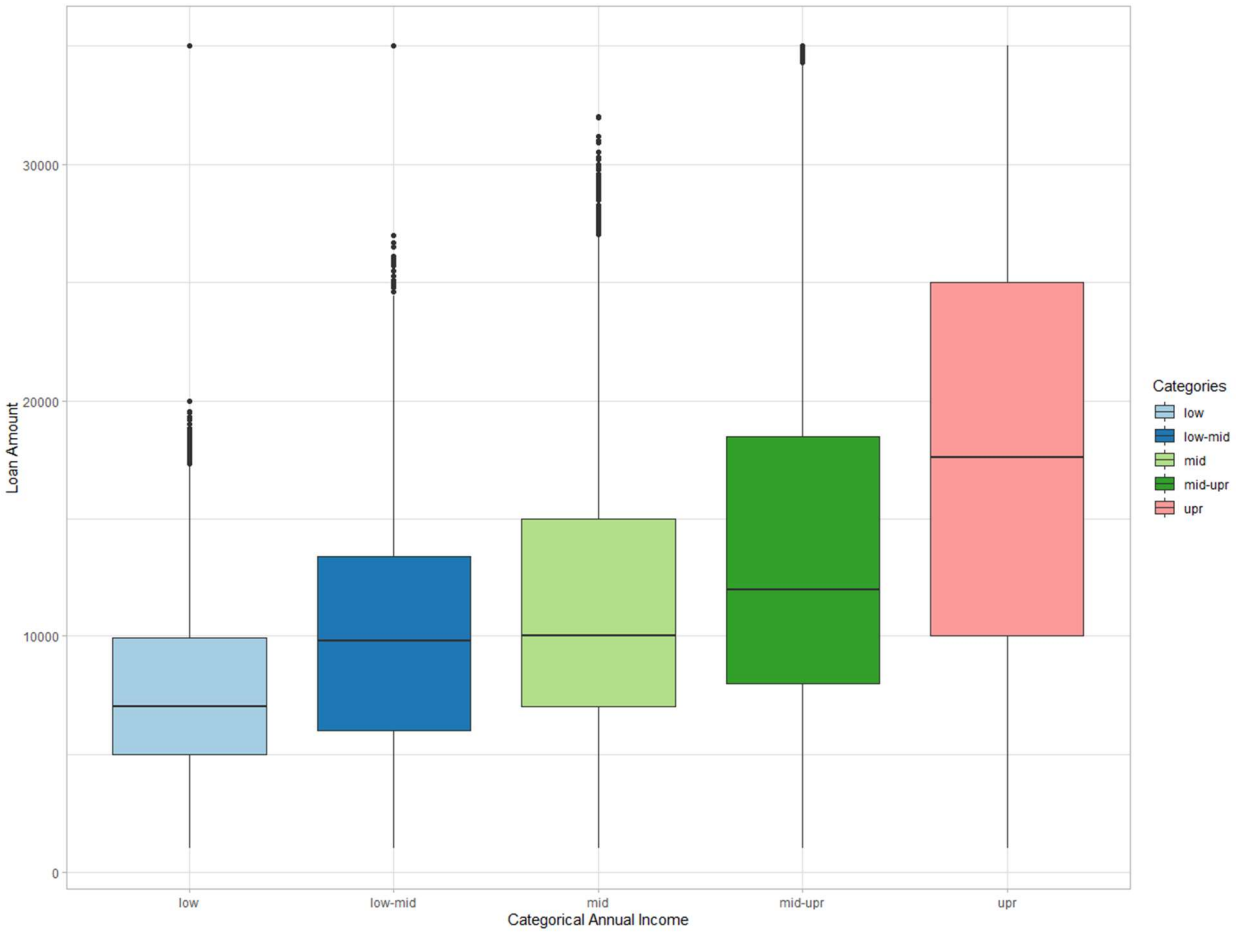
Within each specific purpose, the percentage stays the same over the course of all job periods.

vii-e: emp_length vs annRet



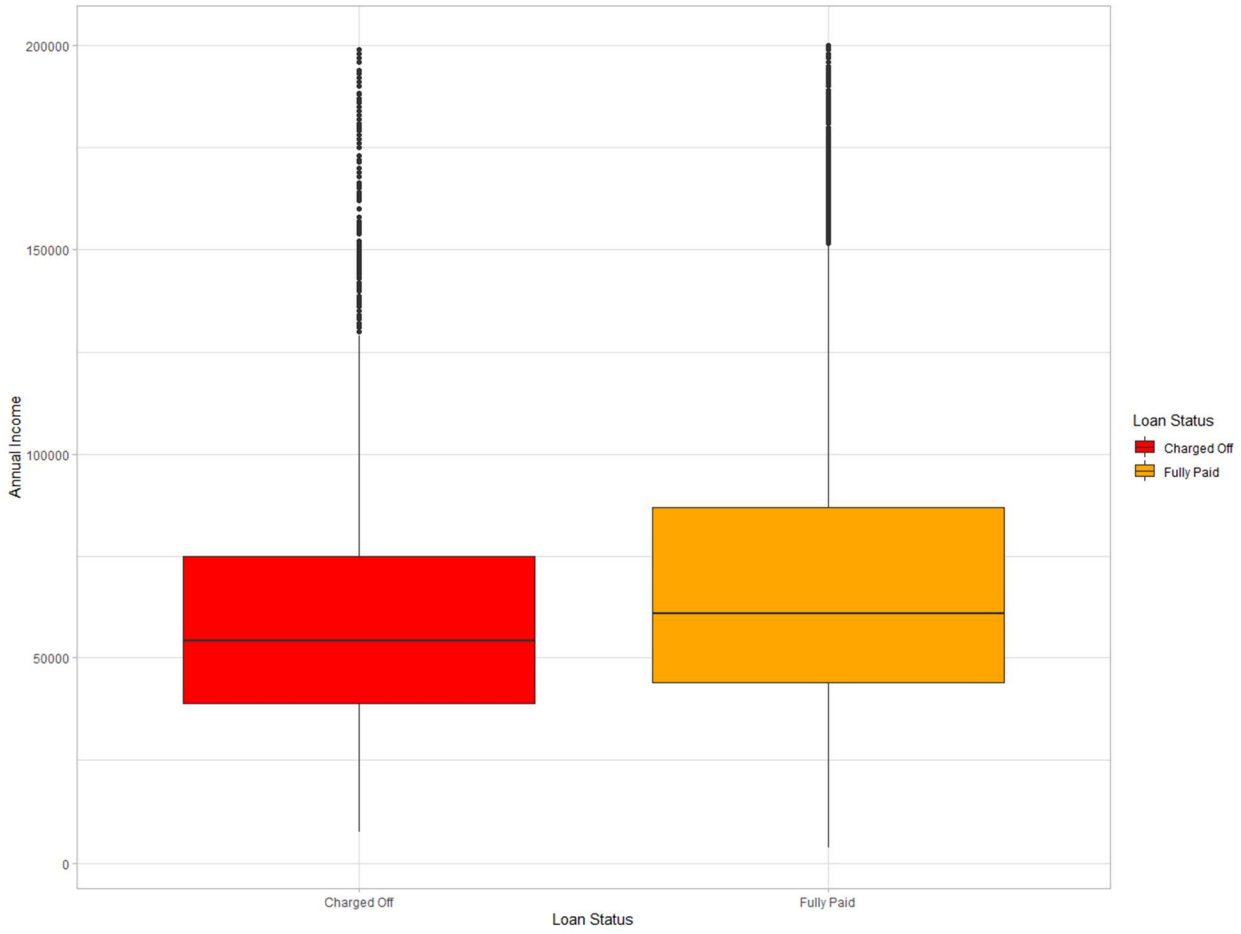
Annual returns are consistent across all lengths of employment.

vii-f: annual_inc vs loan_amnt



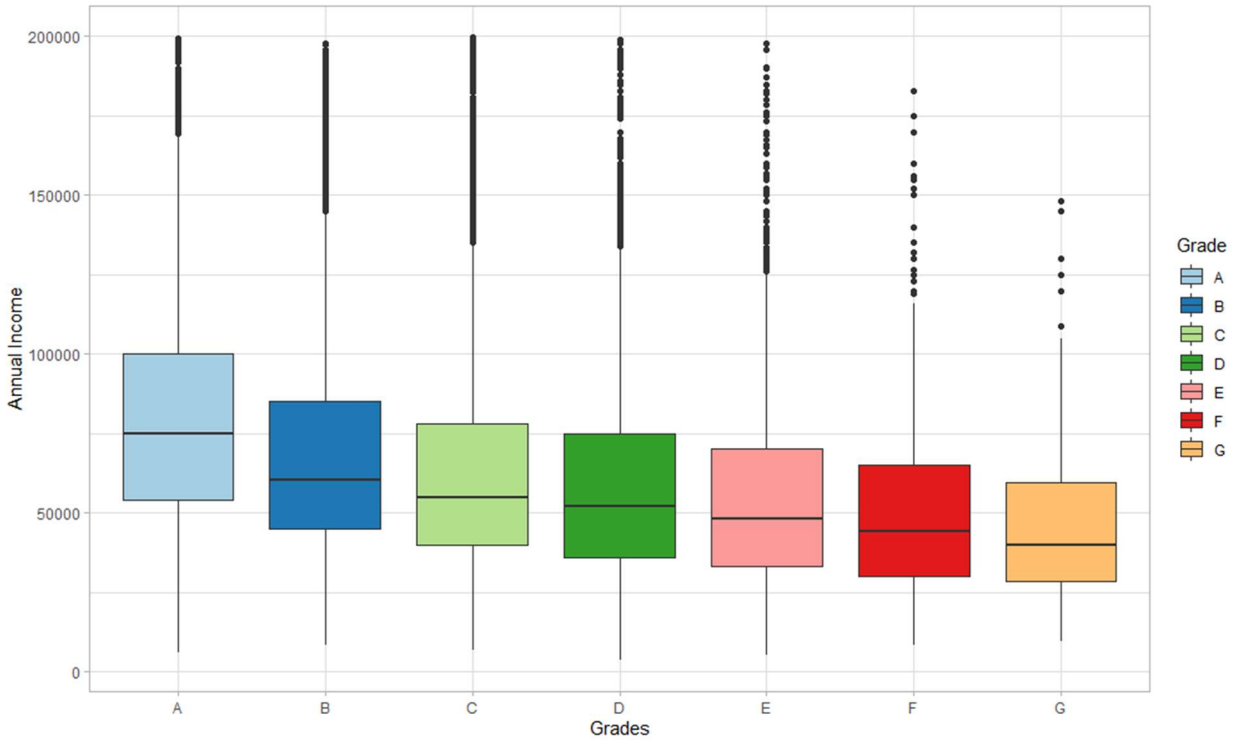
As your annual income rises, so does the loan amount.

vii-g: annual_inc vs loan_status



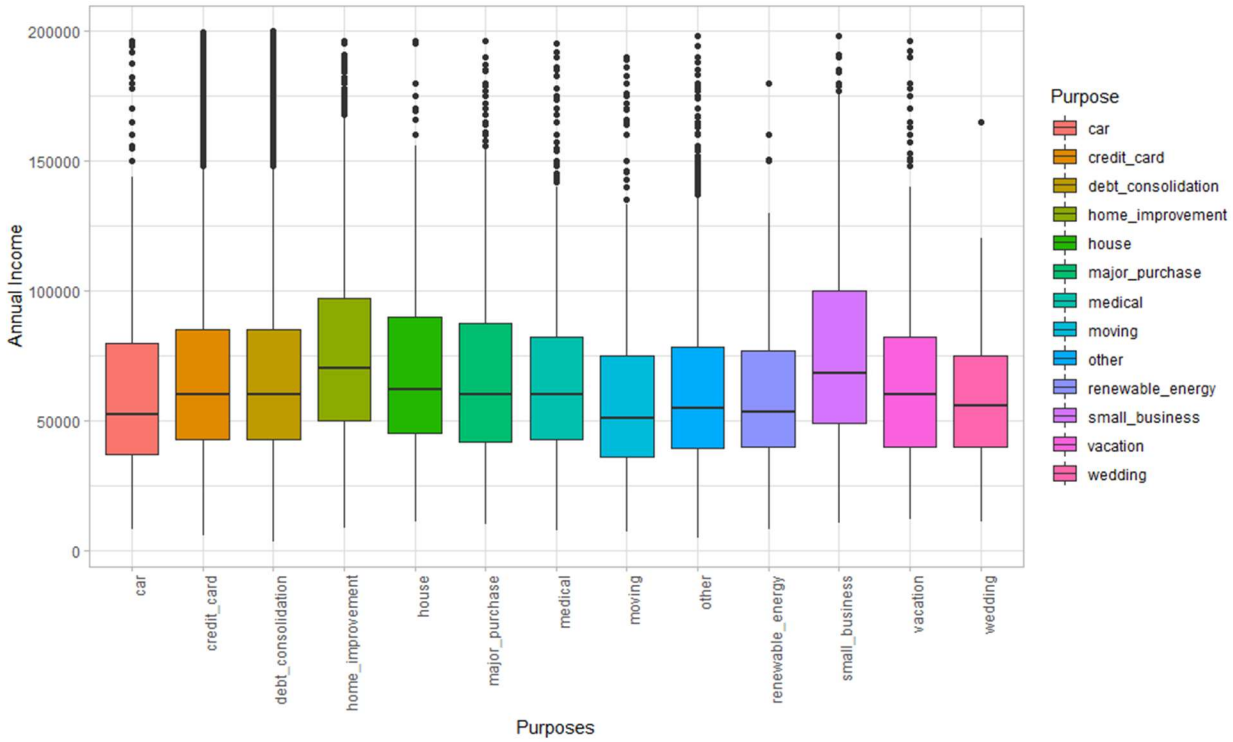
The likelihood that the loan will be repaid increases with annual income.

vii-h: annual_inc vs grade



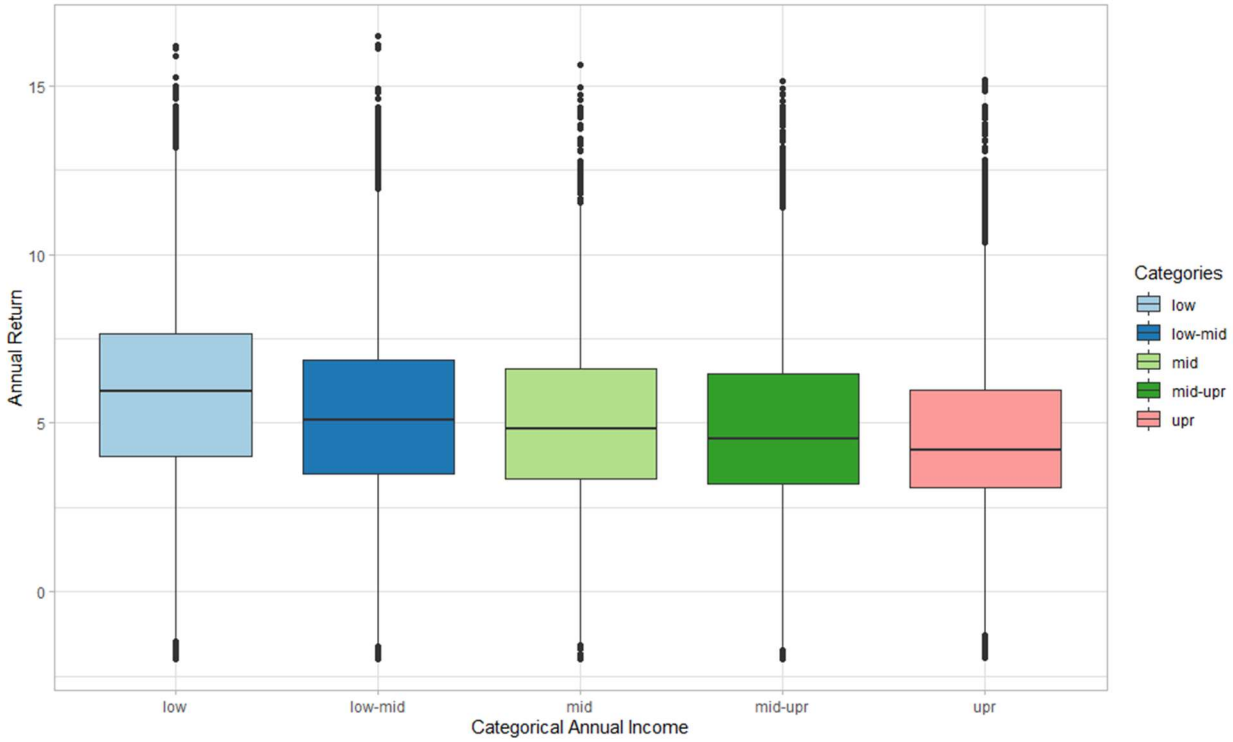
With a higher annual income, you have a better chance of receiving a higher grade.

vii-i: annual_inc vs purpose



The association between purpose and annual income isn't very significant.

vii-j: categorised_annual_inc vs ann_rtn

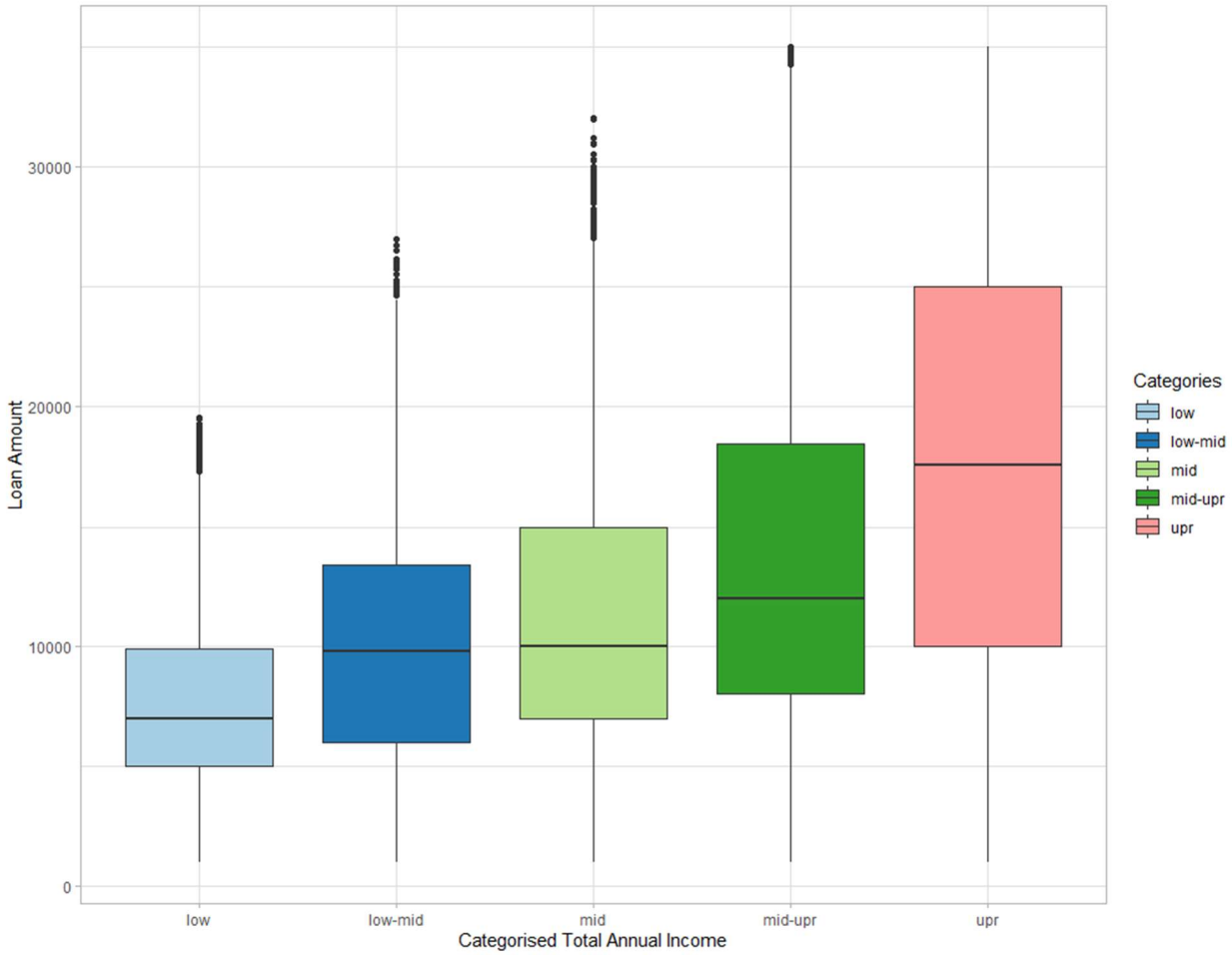


The annual return decreases as income categories go from low to high.

(vii) Generate some (at least 3) new derived attributes which you think may be useful for predicting default., and explain what these are. For these, do an analysis as in the questions above (as reasonable based on the derived variables).

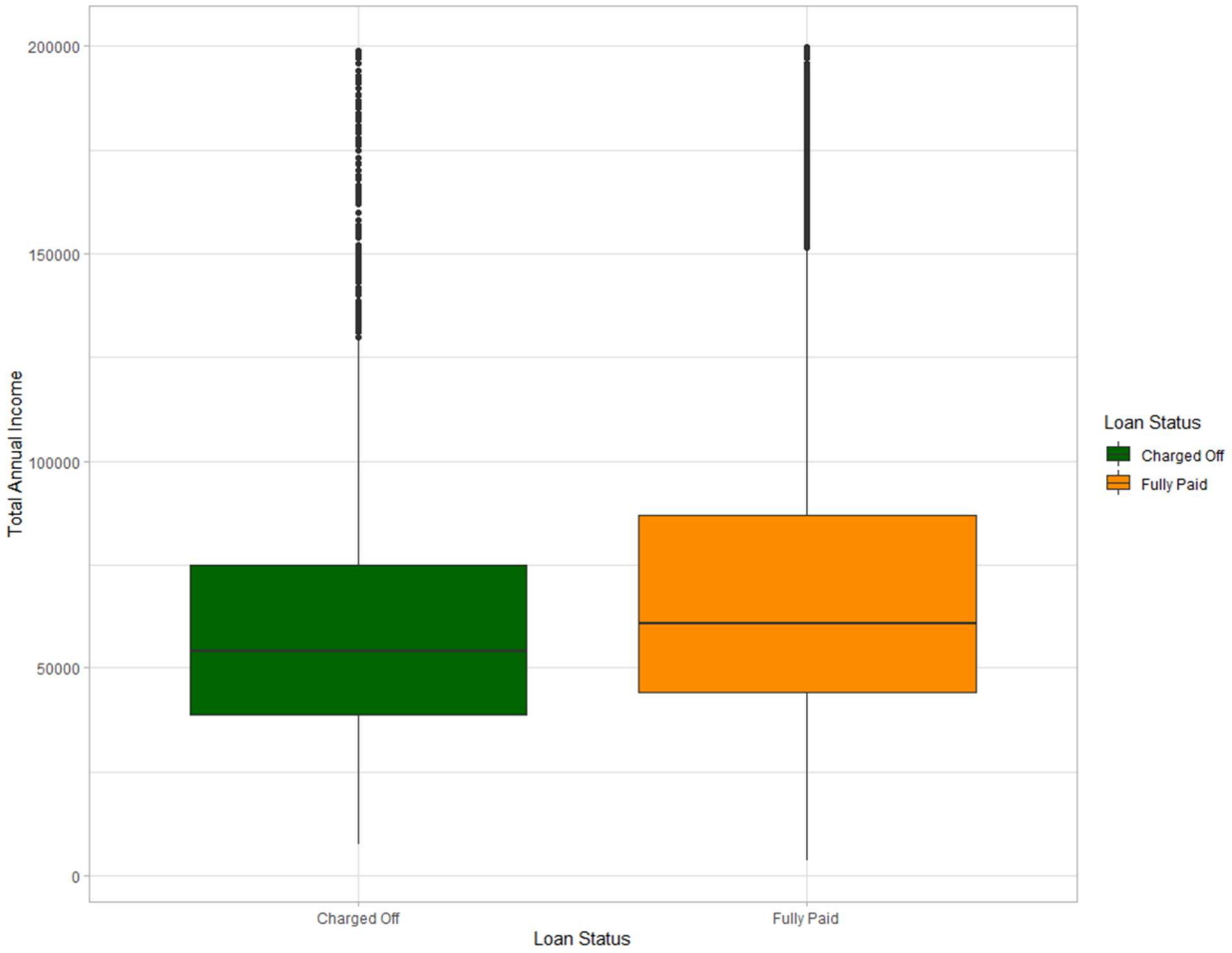
• The combined yearly income of the applicants and co-applicants is the total annual income.

viii-a: categorised_total_annual_inc vs loan_amount



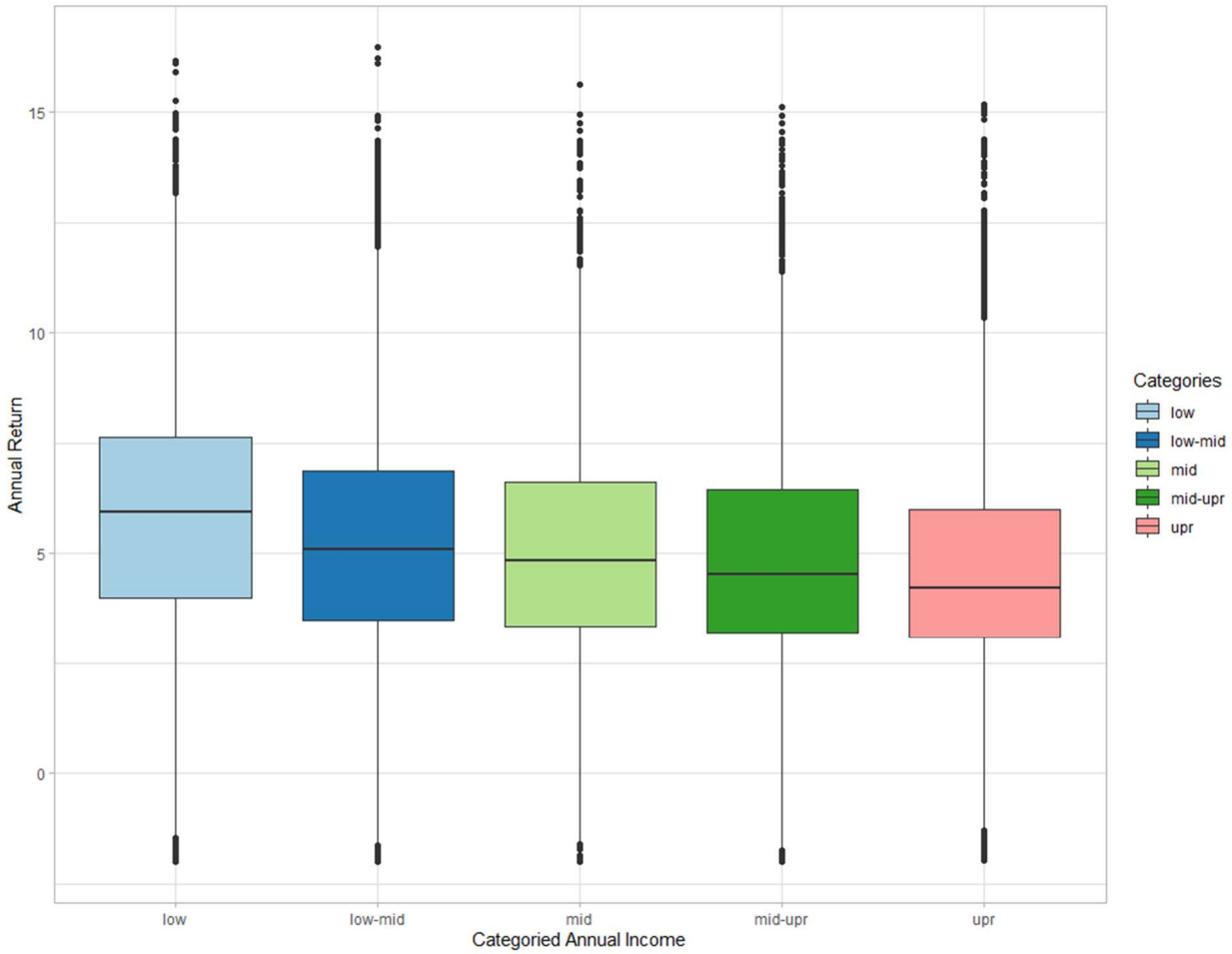
• As Total Annual Income Increases the Loan Amount is also increased

viii-b: loan_status vs total_annual income



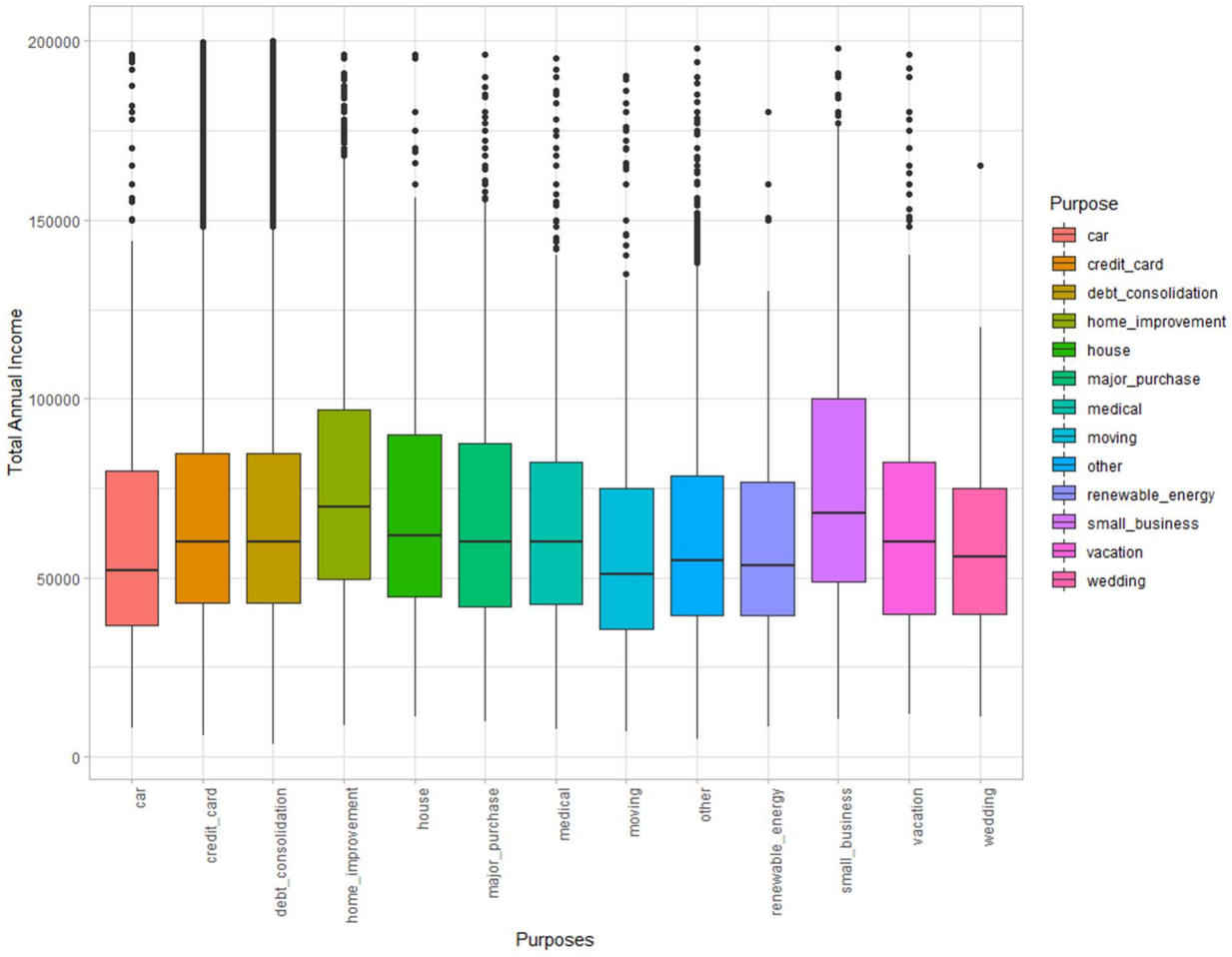
• For Fully Paid loans the total annual income is higher.

viii-c: categorised_annual_income vs annual_return

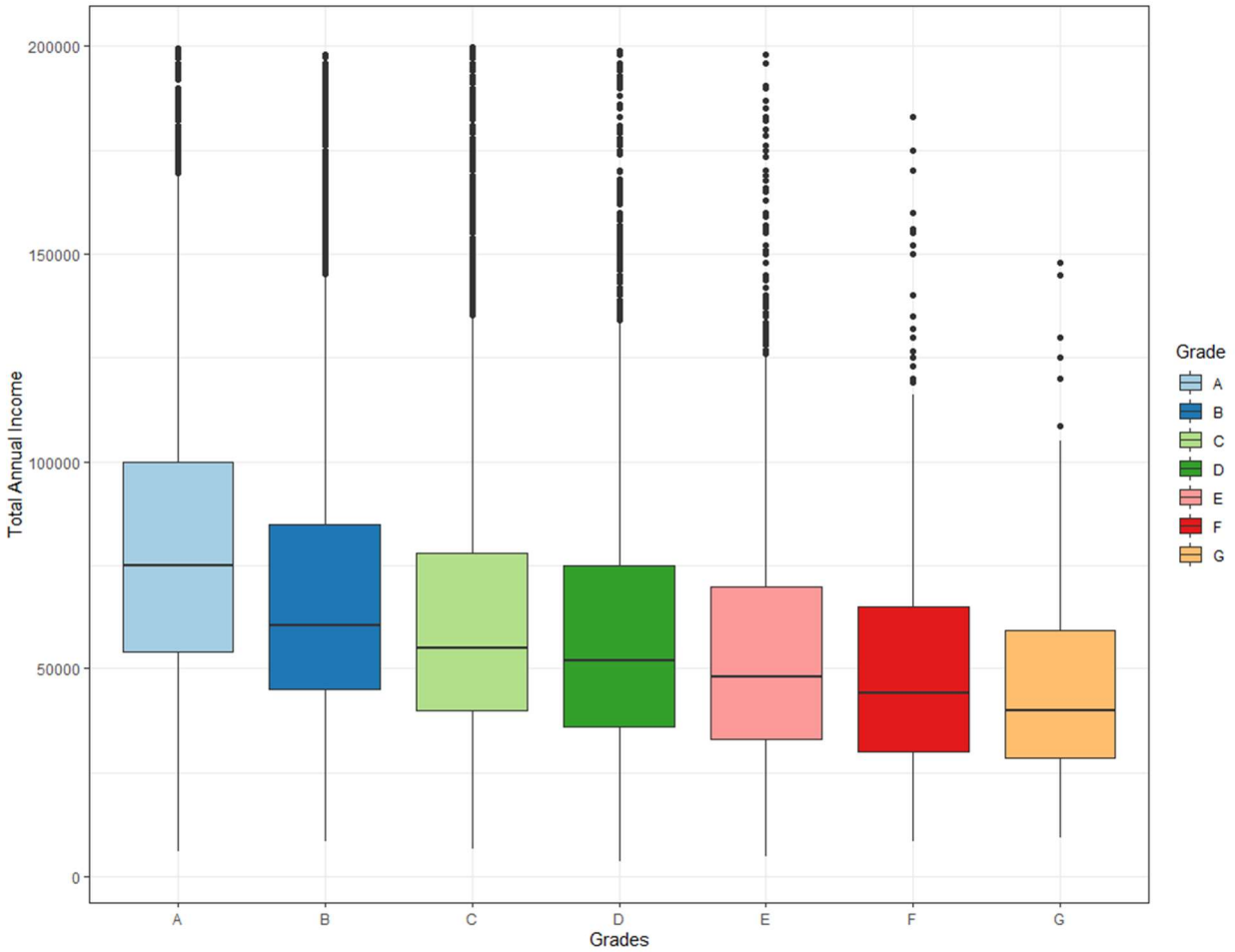


• With Upper annual income the annual income return is decreasing.

viii-d: purposes vs total_annual_income



viii-e: grades vs total_annual_income



• Total annual income decreased with Loan grades.

(b) Are there missing values? What is the proportion of missing values in different variables? Explain how you will handle missing values for different variables. You should consider what the variable is about, and what missing values may arise from – for example, a variable monthsSinceLastDeliquency may have no value for someone who has not yet had a delinquency; what is a sensible value to replace the missing values in this case? Are there some variables you will exclude from your model due to missing values?

We eliminated the columns below since they were absolutely empty.

```
> colnames(lcdf[,missing_col])
 [1] "id"                                "member_id"                         "url"                               "desc"
 [5] "next_pymnt_d"                      "revol_bal_joint"                   "sec_app_earliest_cr_line"          "sec_app_inq_last_6mths"
 [9] "sec_app_mort_acc"                  "sec_app_open_acc"                  "sec_app_revol_util"                "sec_app_open_act_il"
[13] "sec_app_num_rev_accts"             "sec_app_chargeoff_within_12_mths"  "sec_app_collections_12_mths_ex_med" "sec_app_mths_since_last_major_derog"
[17] "hardship_type"                     "hardship_reason"                   "hardship_status"                   "deferral_term"
[21] "hardship_amount"                   "hardship_start_date"               "hardship_end_date"                 "payment_plan_start_date"
[25] "hardship_length"                   "hardship_loan_status"              "orig_projected_additional_accrued_interest" "hardship_payoff_balance_amount"
[29] "hardship_last_payment_amount"      "debt_settlement_flag_date"         "settlement_status"                 "settlement_date"
[33] "settlement_amount"                 "settlement_percentage"
```

The missing value treatment is detailed in the table below.

Col_Name	Replacement/Action	Possible Reason
Verification_status		
dti	We can replace with 0	These are due to individual application
Annual_inc		
revol_util	if borrower has other credit lines, we can replace it by ration of revol_bal and total_rev_hi_lim	Possibly due to borrower doesn't have any other credit line
Emp_title	We can keep column as it is or drop it, as it won't contribute for model prediction	Missing due to information not captured

open_acc_6m	We can drop these columns	This might be due to applicant doesn't have any active installments currently
total_acc		
max_bal_bc…		
inq_last_12m…		
open_il_12m…		
all_util…		
open_il_24m…		
open_rv_12m…		
inq_fi		
Acc_open_past_24mths		
total_cu_tl…		
Mo_sin_rcnt_tl		
il_util…		
open_acc_6m		

mths_since_last_record …	Replacement not required or replace with any high value for example 1000.	Not applicable for someone who has not yet had a public record
mths_since_recent_bc_dlq…	Replacement not required or replace with any high value for example 1000.	Not applicable for someone who has not yet had a delinquency
mths_since_last_major_derog…	Replacement not required or replace with any high value for example 1000.	Not applicable for someone who has not yet had a major derogatory
mths_since_recent_revol_delinq…	Replacement not required or replace with any high value for example 1000.	Not applicable for someone who has not yet had a delinquency
mths_since_last_delinq …	Replacement not required or replace with any high value for example 1000.	Not applicable for someone who has not yet had a delinquency
mths_since_recent_inq	We can drop these columns or put 0	Not applicable for someone who has not made any recent inquiry
num_accts_ever_120_pd	We can drop these columns or put 0	Not applicable for someone who have not 120 days past due

mo_sin_old_il_acct	Replacement not required or replace with any high value for example 1000.	Not applicable for someone who is taking loan first time
bc_util	We can take Ratio of total current balance to high credit/credit limit for all bankcard accounts.	Missing due to information not captured
percent_bc_gt_75…	Replacement not required	Not applicable for someone who have not crossed 75% of limit
bc_open_to_buy	We can replace with median value of all records	Missing due to information not captured
mths_since_recent_bc	Replacement not required or replace with any high value for example 1000.	Not applicable for someone who have not opened bankcard account
last_pymnt_d	Replacement not required due to data leakage	Not applicable for someone who have not initiated first payment
pct_tl_nvr_dlq…	We can replace with average value of all records	Missing due to information not captured

title	Replacement not required or we can drop this record from dataset	Missing due to information not captured
last_credit_pull_d…	Replacement not required or we can drop this record from dataset	Not applicable for someone who have not yet pulled credit
avg_cur_bal	Replacement not required or we can drop this record from dataset	Missing due to information not captured
num_rev_accts…	Replacement not required or we can drop this record from dataset	Not applicable for someone who do not have revolving account

(c) Consider the potential for data leakage. You do not want to include variables in your model which may not be available when applying the model; that is, some data may not be available for new loans before they are funded. Leakage may also arise from variables in the data which may have been updated during the loan period (ie., after the loan is funded). Identify and explain which variables you will exclude from the model.

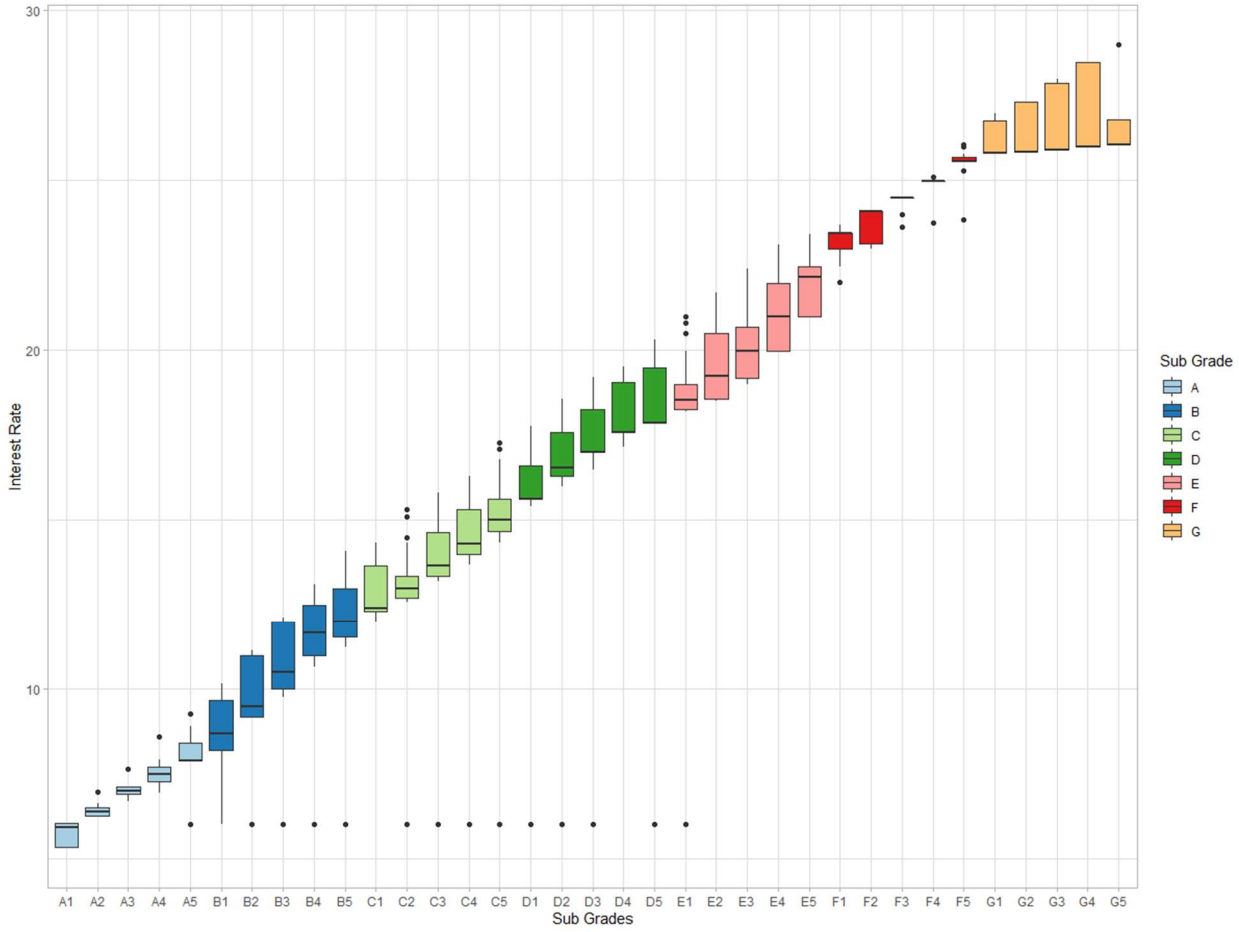
• Data leakage is when variables and information from outside the dataset is used to create a model. It is extremely important to prevent data leakage as it has the potential to ruin your model due to inaccuracy. In this case the variables that need to be considered in order to exclude from the model are variables that are updated during the loan period and variables that were not available when applying the model. The following are only few variables that are listed for the purpose of understanding opposed to every single variable that wouldn't fit the criteria

• The purpose or goal is to develop a model to predict loan default and then decide which loans to invest in so therefore it wouldn't be beneficial to include every single variable. Some of the

variables that shouldn't be included are the ones that changed during and before the loan period, the following are just few examples of those type of variables such as "issue_d, last_pymnt_amnt", "emp_title", "last_pymnt_d", and "acc_open_past_24mths". "Last_pymnt_amt" and "Last_pymnt_d" are both variables that were received before and already paid for. "acc_open_past_24mths" are the trades that were opened in the duration of 2 years before the loans. Lastly "issue_d" is the month in which the loan was funded which no longer would be relevant.

• Other variables that need to be removed are the ones that were revealed after the loan has been issued such as " Collection_recovery_fee", "next_pymnt_d", and "collection_recovery_fee". These are all dependent variables.

• There are also other variables that are unnecessary regardless If it was during or after the loan such as "zip_code", "addr_state", and "emp title".



grade	avgIR	sdIR	minIR	maxIR
A	7.173848	0.966966	5.32	9.25
B	10.75356	1.443157	6	14.09
C	13.84776	1.185915	6	17.27
D	17.19058	1.222019	6	20.31
E	19.92766	1.375556	6	23.4
F	23.98044	0.916387	21.99	26.06
G	26.42563	0.849077	25.8	28.99

sub_grade	avgIR	sdIR	minIR	maxIR
A1	5.680069	0.347485	5.32	6.03
A2	6.415494	0.166259	6.24	6.97
A3	7.094107	0.324701	6.68	7.62
A4	7.475851	0.357395	6.92	8.6
A5	8.241788	0.424467	6	9.25
B1	8.87001	0.721752	6	10.16
B2	9.959382	0.815586	6	11.14
B3	10.84593	0.887329	6	12.12
B4	11.73146	0.839794	6	13.11
B5	12.22738	0.851215	6	14.09
C1	12.86153	0.786176	11.99	14.33
C2	13.3082	0.873285	6	15.31
C3	13.97528	0.865608	6	15.8
C4	14.56803	0.854714	6	16.29
C5	15.22136	0.883442	6	17.27
D1	16.09891	0.870686	6	17.77
D2	16.95641	0.886628	6	18.55
D3	17.44531	0.873474	6	19.2
D4	18.07453	0.831805	17.14	19.52
D5	18.48426	1.002095	6	20.31
E1	18.97299	0.98727	6	21
E2	19.57885	1.058906	18.49	21.7
E3	20.14332	1.032144	18.99	22.4
E4	20.99339	0.952378	19.99	23.1
E5	21.97003	0.762833	20.99	23.4
F1	23.12476	0.59623	21.99	23.7
F2	23.74262	0.47617	22.99	24.08
F3	24.38534	0.247137	23.63	24.5
F4	24.95299	0.214472	23.76	25.09
F5	25.59545	0.272905	23.83	26.06
G1	26.12	0.472927	25.8	26.99
G2	26.39381	0.736468	25.83	27.31
G3	26.73368	1.016706	25.89	27.99
G4	26.99	1.369306	25.99	28.49
G5	26.7925	1.465	26.06	28.99

3. Do a univariate analysis to determine which variables (from amongst those you decide to consider for the next stage prediction task) will be individually useful for predicting the dependent variable (loan_status). For this, you need a measure of relationship between the dependent variable and each of the potential predictor variables. Given loan-status as a binary dependent variable, which measure will you use? From your analyses using this measure, which variables do you think will be useful for predicting loan_status? (Note – if certain variables on their own are highly predictive of the outcome, it is good to ask if this variable has a leakage issue).

Emp_length

```
> lcdf<-lcdf %>% mutate(e_emp_length = case_when(
+   emp_length == '< 1 year' ~0.5,
+   emp_length == '1 year'   ~1.0,
+   emp_length == '10+ years'~10.0,
+   emp_length == '2 years'  ~2.0,
+   emp_length == '3 years'  ~3.0,
+   emp_length == '4 years'  ~4.0,
+   emp_length == '5 years'  ~5.0,
+   emp_length == '6 years'  ~6.0,
+   emp_length == '7 years'  ~7.0,
+   emp_length == '8 years'  ~8.0,
+   emp_length == '9 years'  ~9.0,
+   is.na(emp_length) ~0.0))
> |
```

E_grade

```
> lcdf<-lcdf %>% mutate(e_grade = case_when(
+   grade == 'G' ~1.0,
+   grade == 'F' ~2.0,
+   grade == 'E' ~3.0,
+   grade == 'D' ~4.0,
+   grade == 'C' ~5.0,
+   grade == 'B' ~6.0,
+   grade == 'A' ~7.0))
> |
```

E_home_ownership

```
> lcdf$e_emp_length[is.na(lcdf$e_grade)] <- 0
> #One hot encoding for home_ownership
> lcdf<-lcdf %>% mutate(e_home_ownership = case_when(
+   home_ownership == 'RENT' ~1.0,
+   home_ownership == 'MORTGAGE' ~3.0,
+   home_ownership == 'OWN' ~5.0))
> |
```

Verification_status

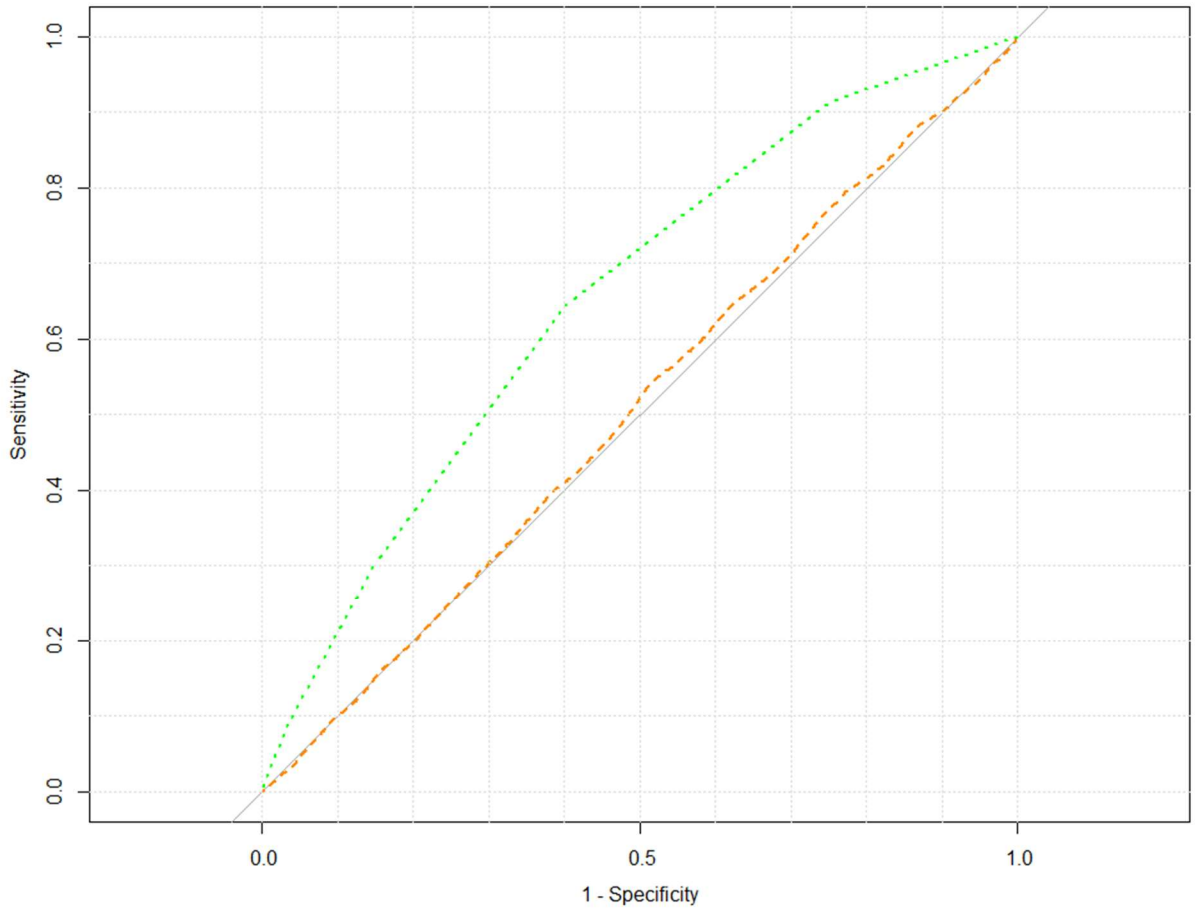
```
> as.data.frame(table(lcdf$verification_status))
             Var1  Freq
1    Not Verified 33202
2 Source Verified 36628
3        Verified 30170
> lcdf<-lcdf %>% mutate(e_verification_status = case_when(
+   verification_status == 'Not Verified' ~1.0,
+   verification_status == 'Source Verified' ~3.0,
+   verification_status == 'Verified' ~5.0))
> # replacing NA to 0
> lcdf$e_emp_length[is.na(lcdf$mths_since_last_delinq)] <- 0
> lcdf$e_emp_length[is.na(lcdf$avg_cur_bal)] <- 0
> lcdf$e_emp_length[is.na(lcdf$mths_since_last_record)] <- 0
> lcdf$e_emp_length[is.na(lcdf$revol_util)] <- 0
> lcdf$e_emp_length[is.na(lcdf$mths_since_last_delinq)] <- 0
> lcdf$e_emp_length[is.na(lcdf$e_verification_status)] <- 0
>
```

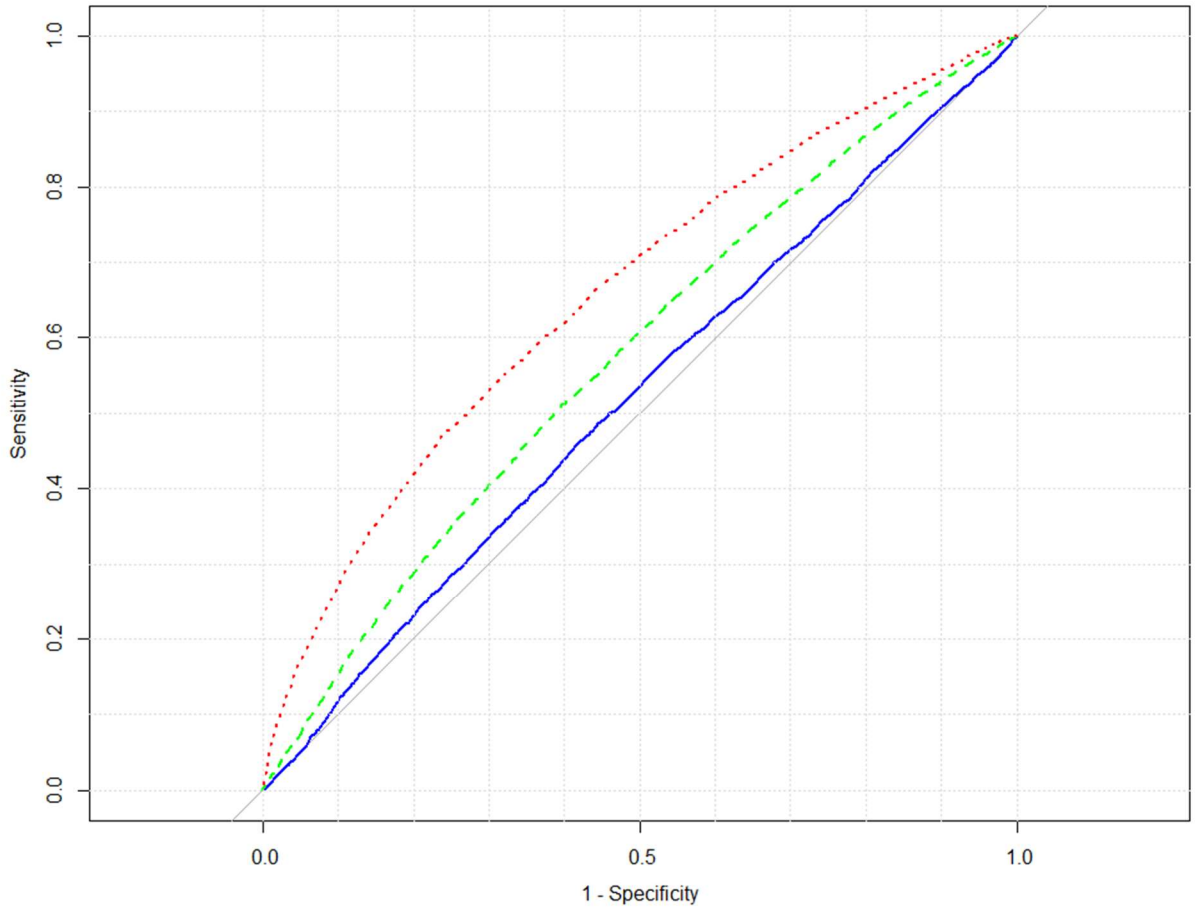
After that, determine the AUC values for each numerical and categorical variable (converted first into numeric values using O-H-E). As shown in the picture below.

```
#Numeric variables:
AUCNum <- sapply(lcdf %>% select_if(is.numeric), auc, response=lcdf$loan_status)
#Numeric and Factor variables:
AUCAll<- sapply(lcdf %>% mutate_if(is.factor, as.numeric) %>% select_if(is.numeric),
                auc, response=lcdf$loan_status)
tidy(AUCAll) %>% view()
#Variables have AUC > 0.5
AUCAll[AUCAll>0.5]
#arranged in descending order to get important variable
tidy(AUCAll) %>% arrange(desc(AUCAll))
# Using tidy(..) function we converts the 'messy' output into a tidy form as a tibble
tidy(AUCAll[AUCAll > 0.5]) %>% view()
auc_call<-tidy(AUCAll[AUCAll > 0.5])
write.csv(auc_call, "auc_call.csv", row.names = FALSE)
```

names	x
loan_amnt	0.521140151
funded_amnt	0.521140151
funded_amnt_inv	0.521147414
int_rate	0.658148292
installment	0.507186537
grade	0.653829826
annual_inc	0.576780409
dti	0.568269628
inq_last_6mths	0.551487208
mths_since_last_deli	0.510867892
open_acc	0.507948233
revol_bal	0.536733235
revol_util	0.531484444
total_acc	0.518490733
total_pymnt	0.755793776
total_pymnt_inv	0.755798683
total_rec_prncp	0.828559597
total_rec_int	0.541562594
recoveries	0.878491114
collection_recovery_	0.859920203
last_pymnt_amnt	0.768416344
tot_cur_bal	0.561194991
open_acc_6m	0.5039498
total_rev_hi_lim	0.565574295
acc_open_past_24mtl	0.582589695
avg_cur_bal	0.569155349
bc_open_to_buy	0.573858463
bc_util	0.543196935
mo_sin_old_il_acct	0.524700765
mo_sin_old_rev_tl_o	0.551115488
mo_sin_rcnt_rev_tl_c	0.553833454
mo_sin_rcnt_tl	0.559670398
mort_acc	0.558319586
mths_since_recent_b	0.553266464
mths_since_recent_i	0.558501969
num_bc_tl	0.51526255
num_il_tl	0.509902095
num_op_rev_tl	0.517655561
num_rev_accts	0.507833293
num_sats	0.507744874
num_tl_120dpd_2m	0.500002643
pct_tl_nvr_dlq	0.512397933
tot_hi_cred_lim	0.573551235
total_bal_ex_mort	0.516919212
total_bc_limit	0.573007896
total_il_high_credit_l	0.511631452
annRet	0.965918399
totAnnInc	0.576803014
e_emp_length	0.534499791
e_grade	0.653829826
e_home_ownership	0.542130482
actualTerm	0.66393319
actualReturn	0.985948677

```
#Now removing data leakage variables
varsOmit<-c("annRet","addr_state","bc_open_to_buy","collection_recovery_fee",
            "collections_12_mths_ex_med","debt_settlement_flag","delinq_amnt",
            "disbursement_method","dti","e_emp_length","e_grade","e_home_ownership",
            "e_verification_status","emp_title","funded_amnt","funded_amnt_inv",
            "hardship_flag","inq_last_6mths","issue_d","last_credit_pull_d",
            "last_pymnt_amnt","last_pymnt_d","mo_sin_rcnt_rev_tl_op",
            "mths_since_recent_bc","num_actv_bc_tl","num_bc_tl","num_rev_accts",
            "num_tl_120dpd_2m","num_tl_op_past_12m","out_prncp","out_prncp_inv",
            "policy_code","pub_rec_bankruptcies","pymnt_plan","recoveries","revol_util",
            "term","title","tot_coll_amt","total_acc","total_bal_ex_mort","total_pymnt",
            "total_pymnt_inv","total_rec_int","total_rec_late_fee","total_rec_prncp",
            "total_rev_hi_lim","zip_code","earliest_cr_line","actualTerm","actualReturn")
```

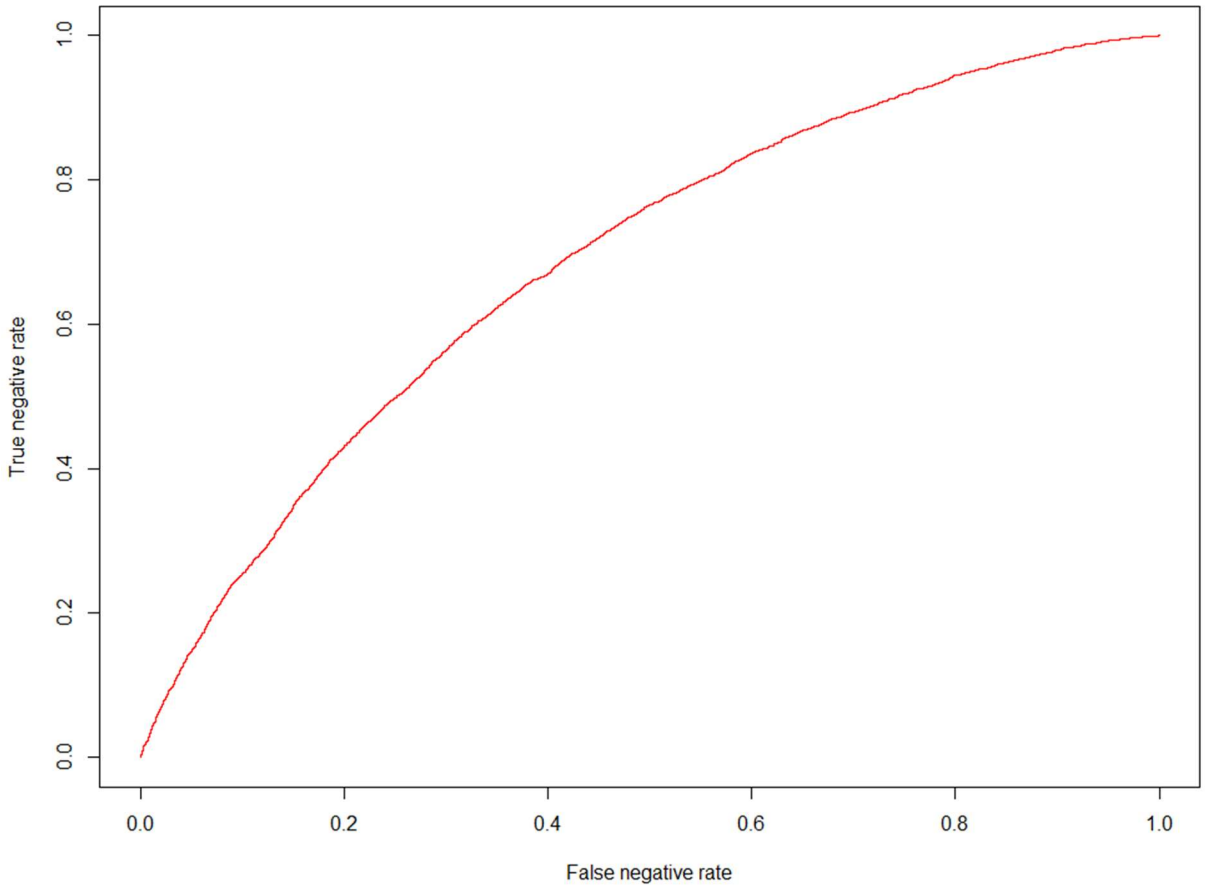


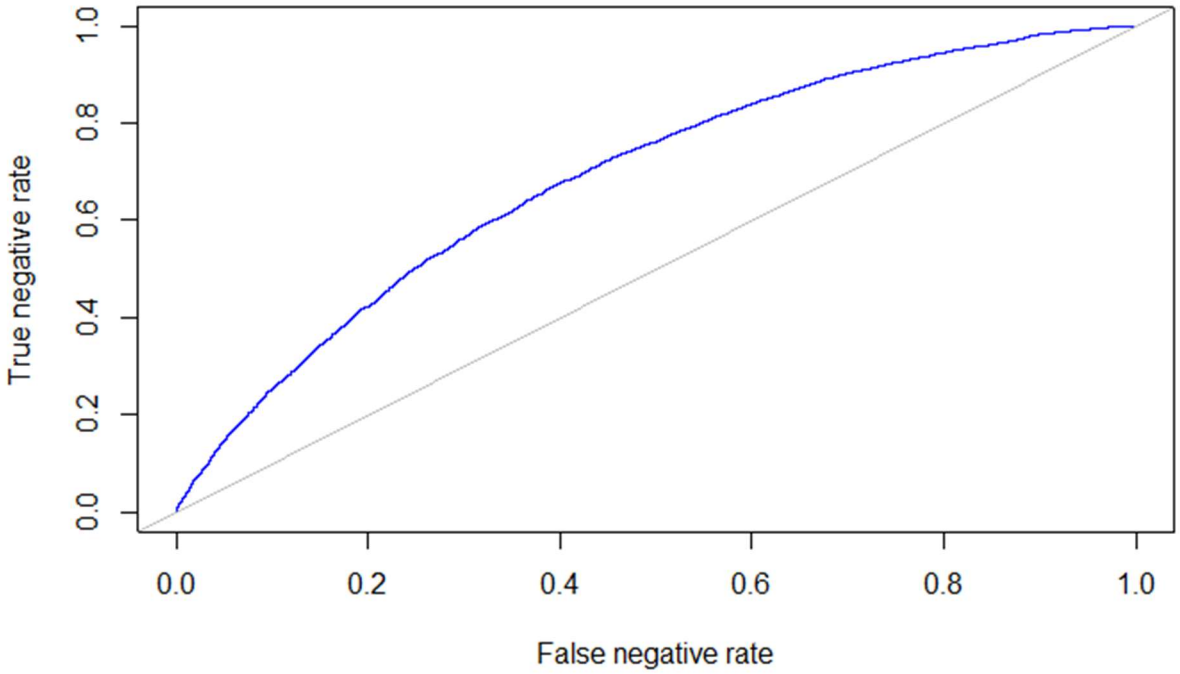


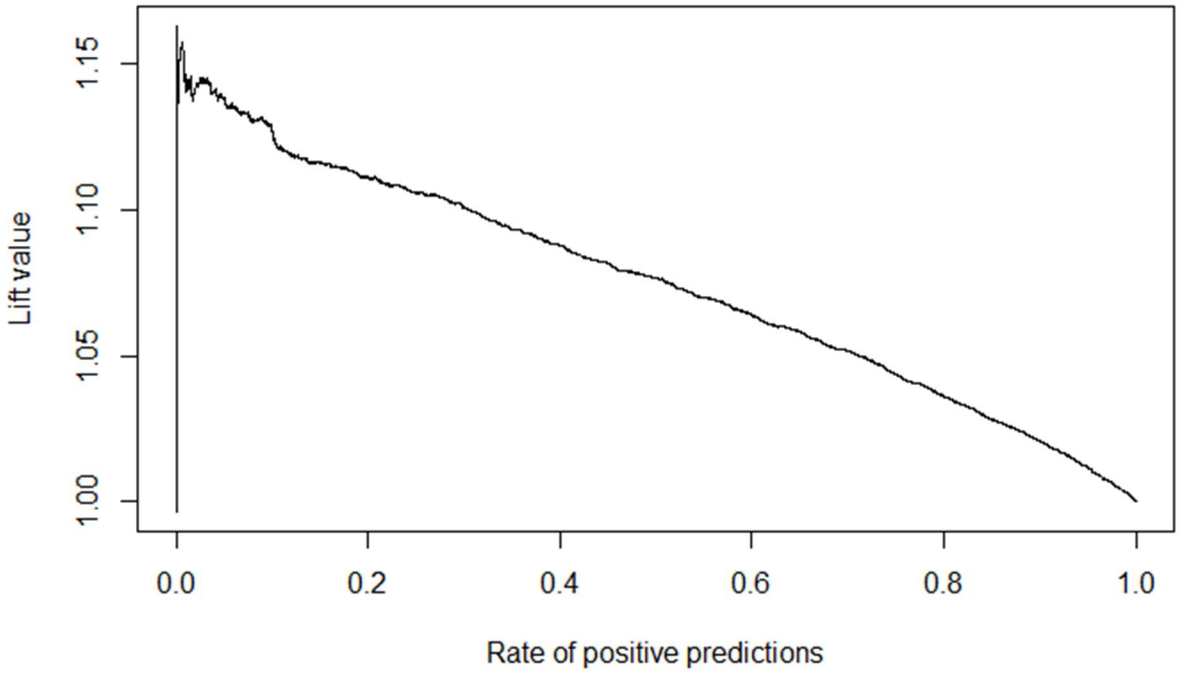
4. (a) Split the data into training and validation sets. What proportions do you consider, why?

We divided the data for the decision tree models into training and validation sets, with the proportion for training sets being 70% and the proportion for validation sets being 30%, because the results for the predictions appeared promising for 70:30 when compared to 50:50 for overall performance. We are now exploring a 70:30 split in light of the strong results.

(b) How will you evaluate performance – which measure do you consider, and why?
For evaluation of models, you should include confusion matrix related measures, as well as ROC analyses and lifts. Explain which performance measures you focus on, and why.


True negative rate
0.0
0.2
0.4
0.6
0.8
1.0
False negative rate
0.0
0.2
0.4
0.6
0.8
1.0





5. Develop a decision tree model to predict default.

Train decision tree models (use either rpart or c50)
What parameters do you experiment with, and what performance do you obtain (on training and validation sets)? Clearly tabulate your results and briefly describe your findings.
[If something looks too good, it may be due to leakage – make sure you address this]
Identify the best tree model. Why do you consider it best? Describe this model – in terms of complexity (size). Examine variable importance. How does this relate to your uni-variate analysis in Question 3 above? Briefly describe how variable importance is obtained (the process used in the decision tree learning algorithm you use(rpart or c50).

Details using rpart

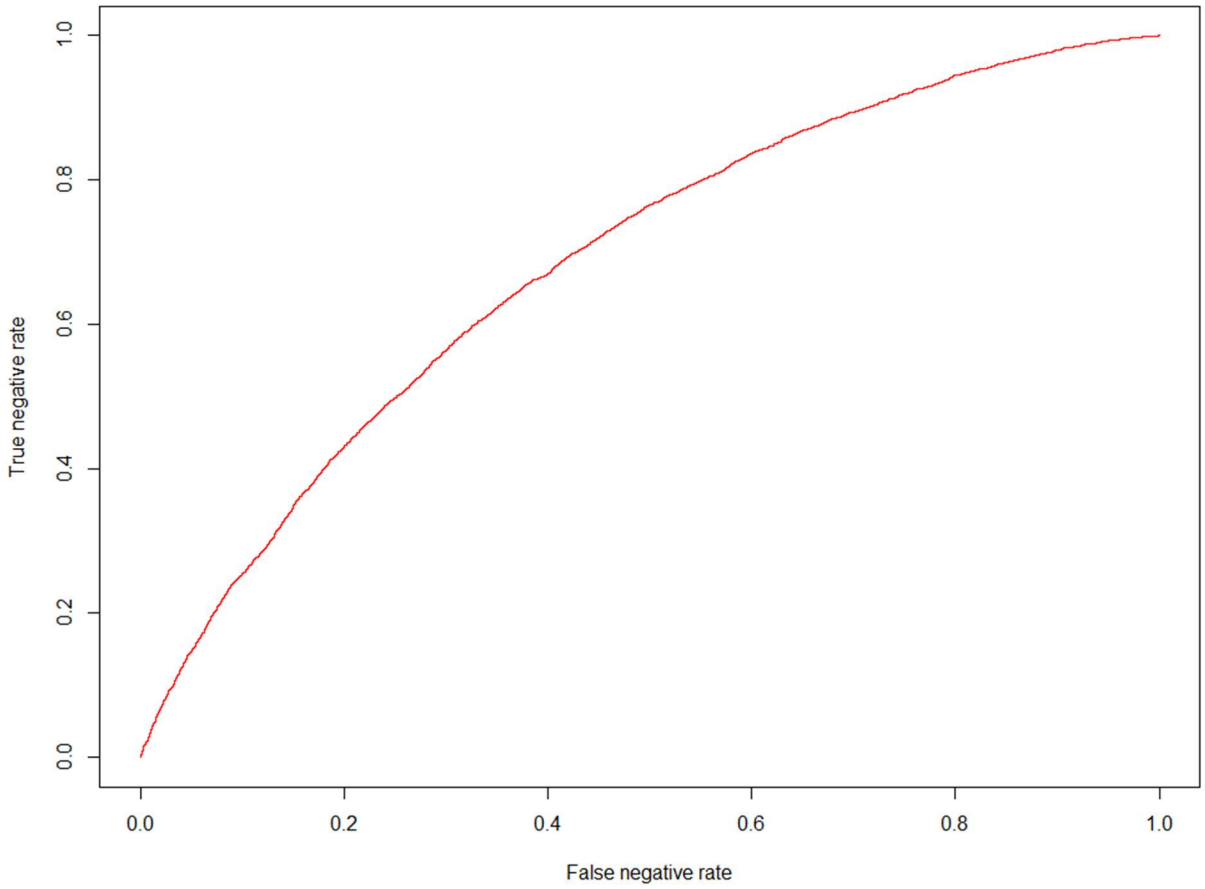
We created a list of factors that contribute to data leaking and used rpart to remove them from the training model. As a result, we can forecast using actual data without adding any further information. The outcome with the data leakage variable is shown in the first table, whereas the outcome without it is shown in the second table.
The table below shows what would happen if the data leak variable wasn't deleted. This has a high degree of accuracy (almost 99.999 percent)

```
> table(pred=predict(DTree1,lcdfTrain, type="class"), true=lcdfTrain$loan_status)
            true
pred          Charged Off Fully Paid
  Charged Off        9655           2
  Fully Paid            0       60343
> |
```

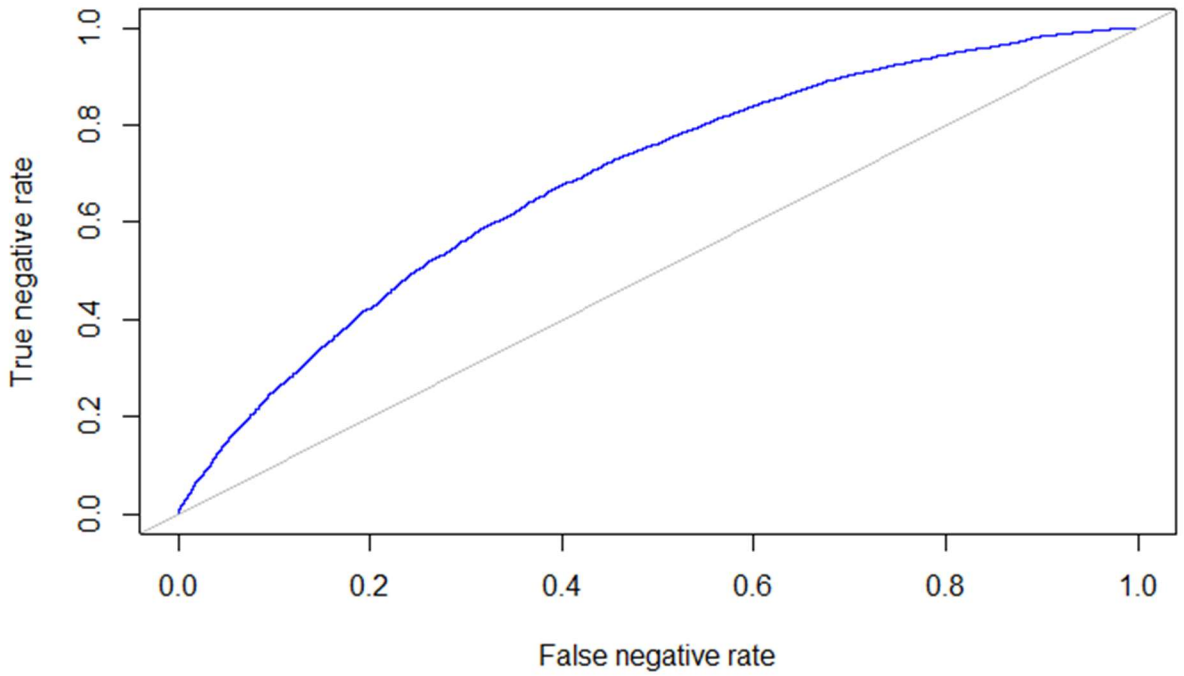
The findings after the data leaking variable was removed are shown in the table below.

```
> table(pred=predict(DTree7,lcdfTrain, type="class"), true=lcdfTrain$loan_status)
            true
pred          Charged Off Fully Paid
  Charged Off        7041           0
  Fully Paid         2590       60369
> table(pred=predict(DTree7,lcdfTest, type="class"), true=lcdfTest$loan_status)
            true
pred          Charged Off Fully Paid
  Charged Off        3035           0
  Fully Paid         1119       25846
> score=predict(DTree7,lcdfTest, type="prob")[,"Charged Off"]
```

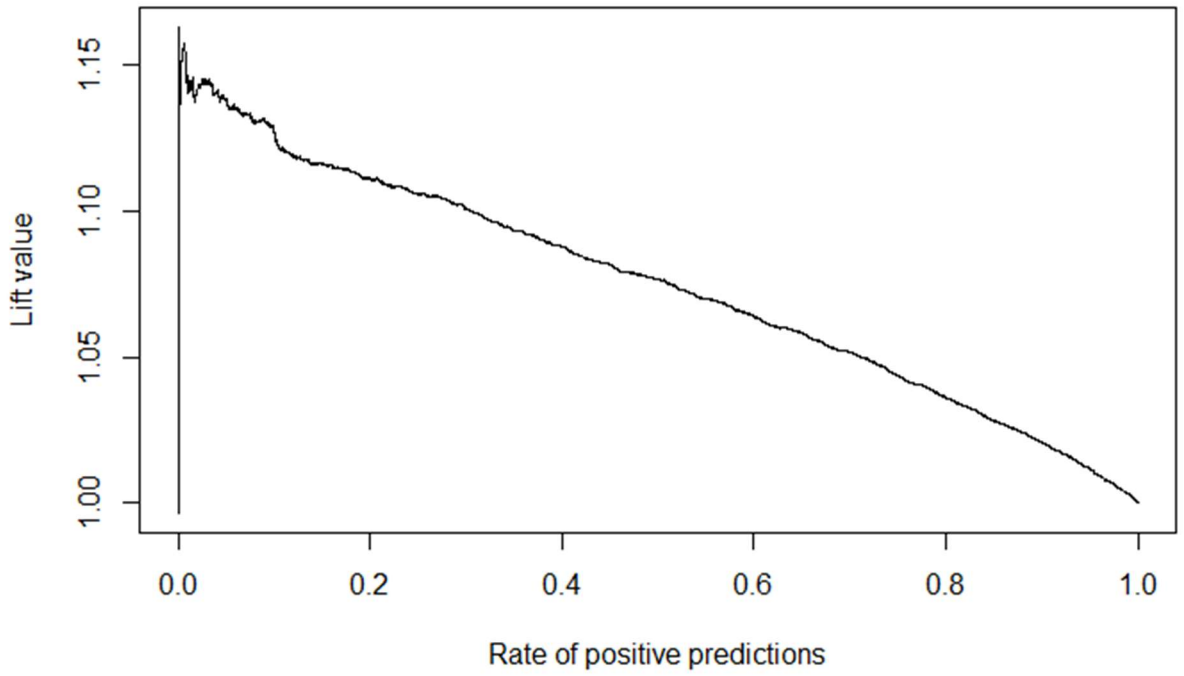
ROC


True negative rate
0.0
0.2
0.4
0.6
0.8
1.0
False negative rate
0.0
0.2
0.4
0.6
0.8
1.0

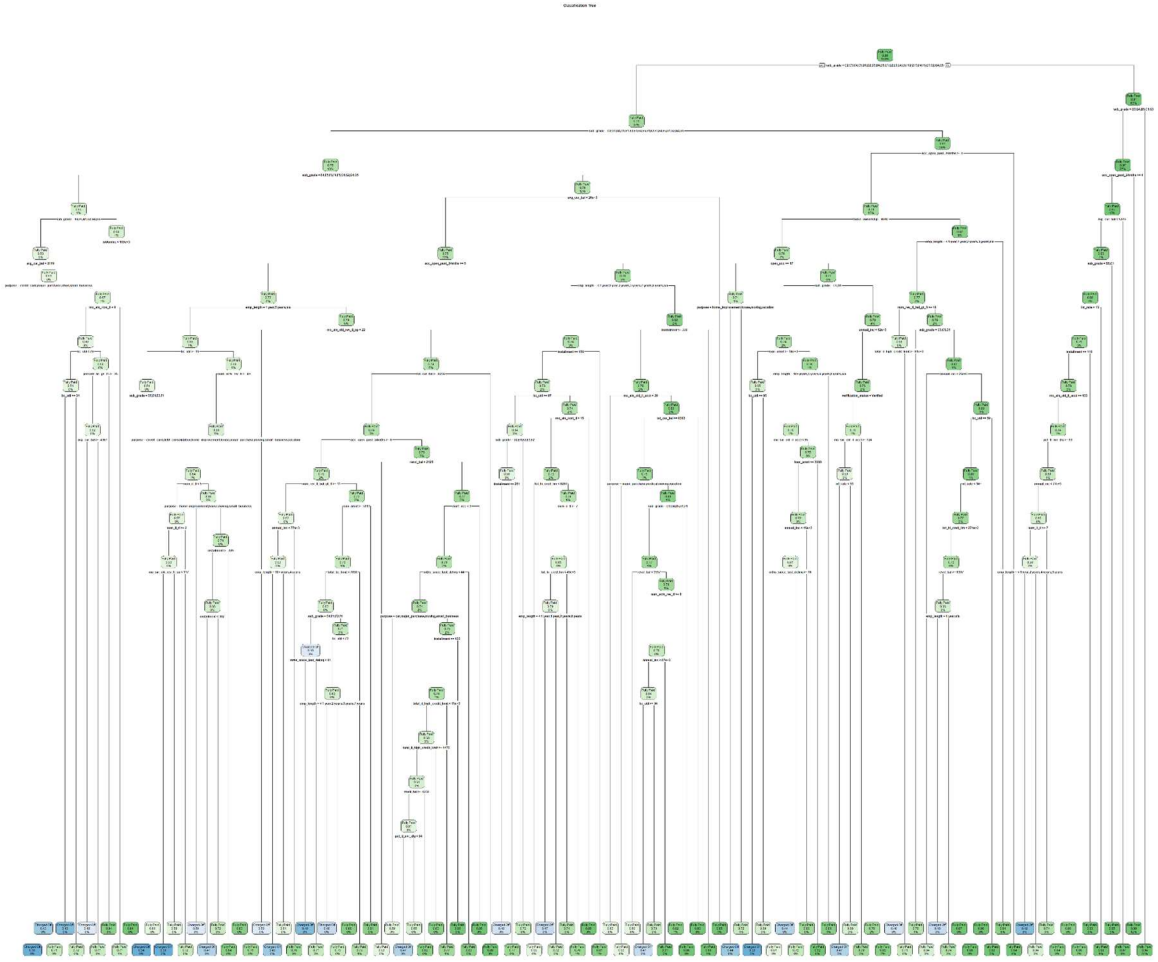
AUC


True negative rate
0.0
0.2
0.4
0.6
0.8
1.0
False negative rate

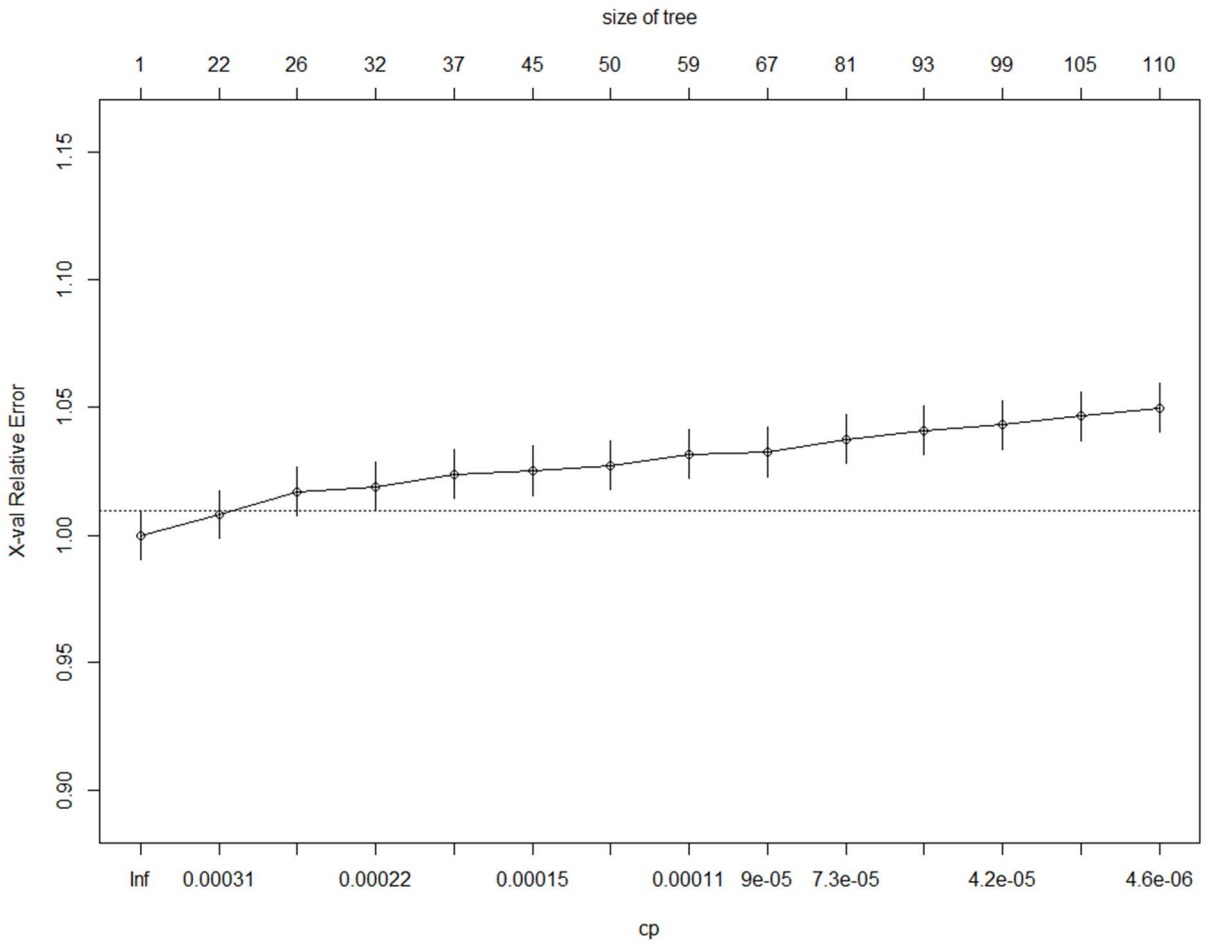
LIFT


Lift value
1.00
1.05
1.10
1.15
0.0
0.2
0.4
0.6
0.8
1.0
Rate of positive predictions

```
> library(ranger)
> DTree12$variable.importance
collection_recovery_fee     debt_settlement_flag
              11450.330                 1711.943
>
```



DECISION TREE



6. (a) Develop random forest and boosted tree model (using gbm or xgb) Note the 'ranger' library and xgb can give faster computations.
What parameters do you experiment with, and how does this affect performance? Describe the best random forest and boosted tree model in terms of number of trees, performance, variable importance.
(b)Compare the performance of random forest, boosted tree and decision tree model from Q 5 above. Do you find the importance of variables to be different ? Which model would you prefer, and why ?

To assess its effectiveness, we have constructed multiple random forest models using ranger and set out all the parameters and possible values for them.

```
library(ranger)
library(xgboost)

#Best Random Forest Model
#Running loop through all possible combination of parameters to obtain best combination of parameter

for (nt in c(500))
{
  for (mt in c(9))
  {
    for (imp in c('permutation'))
    {
      for (pb in c(TRUE))
      {
        for (mns in c(10))
        {
          for (md in c(10))
          {
            for (sr in c("gini"))
            {
              for (rf in c(0.1))
              {
```

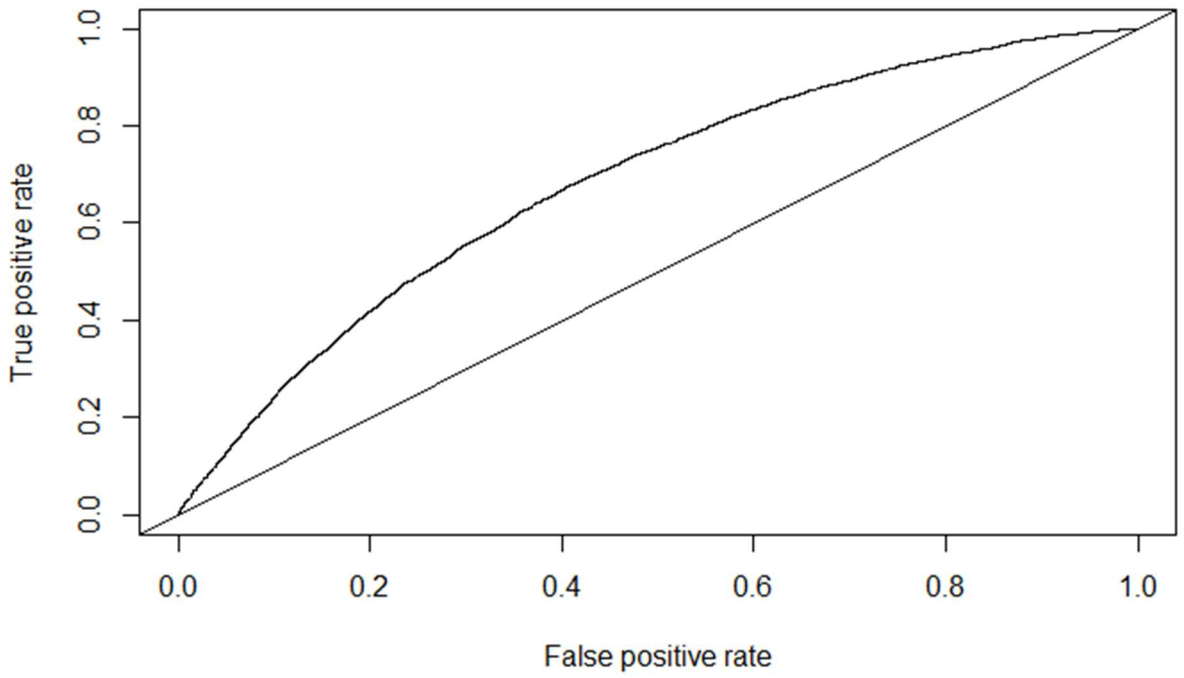
Parameter	Possible values	Effect on Model
num.trees	50,100,200,500	Model performance does improve with the number of trees
mtry	5,6,7,8,9,10,11	We have started with 5 and model performance are getting to increase initially. It was best for 7,8 and 9. However after that model performance was getting poorer. As per document best value is square root of number of variables.
importance	'permutation','impurity'	Initially we worked with permutation. However impurity was giving better AUC for most of the models on test data.
probability	TRUE	Kept same for all model
min.node.size	1,5,8,9,10,11,13	For smaller values of minimum node size, model was not giving good results. It was optimum at 7,9,10. Again dropped with larger values
max.depth	1,2,3,4,5,6,7,8,9,10	Like minimum node size, It was giving better results for 7,9,10.
splitrule	gini, extratrees,variance	We have experimented with other values, but Gini was better one
regularization.factor	0.1,0.3,0.5,0.7,0.9,1.0	We have tried many values between 0.0 and 1.0 and found better results around 0.9 and 1.0

BEST RANDOM FOREST DECISION TREE

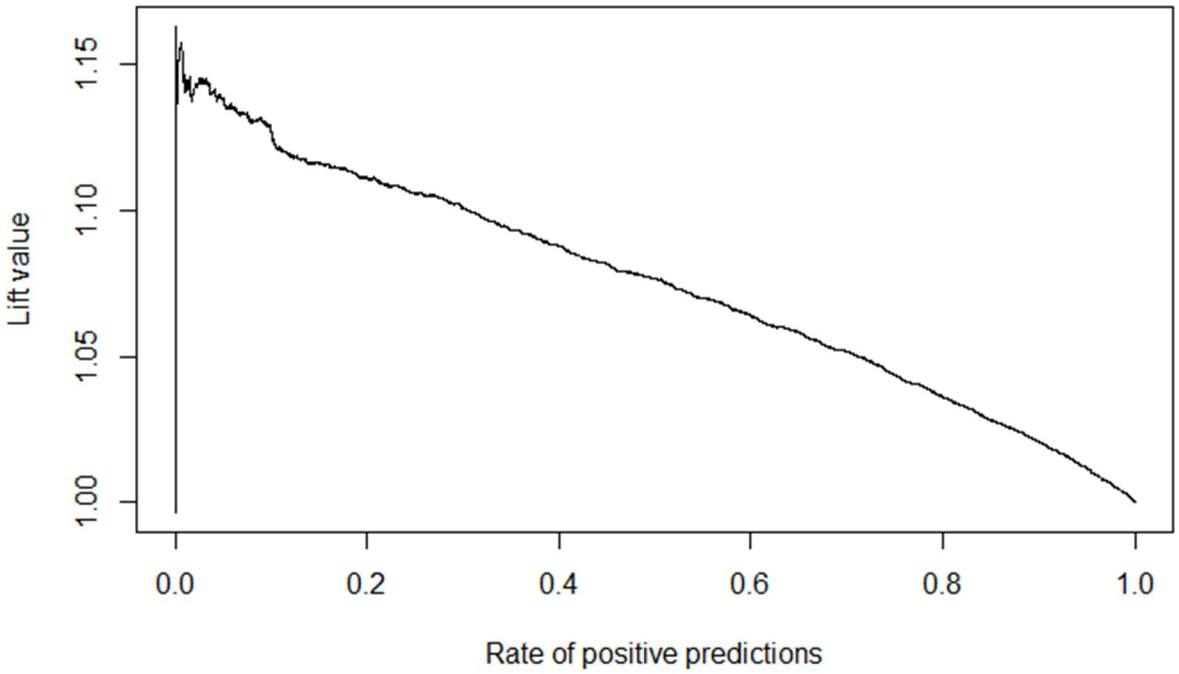
```
[[1]]
[1] 0.6797449

[1] "num_trees: "
[1] 500
[1] "mtry: "
[1] 9
[1] "importance: "
[1] "permutation"
[1] "probability: "
[1] TRUE
[1] "minimum_nod_size: "
[1] 10
[1] "max_depth: "
[1] 10
[1] "splitrule: "
[1] "gini"
[1] "regularization_factor: "
[1] 0.1
[1] "--------------------------------------"
```

ROC



LIFT




```
> confusionMatrix(predData, as.factor(lcdfTrain$loan_sta
Confusion Matrix and Statistics

             Reference
Prediction    Charged Off Fully Paid
  Charged Off         116          0
  Fully Paid         9511      60373

               Accuracy : 0.8641
                 95% CI : (0.8616, 0.8667)
    No Information Rate : 0.8625
    P-Value [Acc > NIR] : 0.1023

                  Kappa : 0.0206

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.012049
            Specificity : 1.000000
         Pos Pred Value : 1.000000
         Neg Pred Value : 0.863903
             Prevalence : 0.137529
         Detection Rate : 0.001657
   Detection Prevalence : 0.001657
      Balanced Accuracy : 0.506025

       'Positive' Class : Charged Off
```

Because it creates a lot of tiny trees to improve model performance, random forest outperforms other forests.

```
> DTree12$variable.importance
          sub_grade           int_rate              grade acc_open_past_24mths         emp_length          loan_amnt
        677.3462978        544.4601687        489.3436943         86.5079898         84.2990577         74.5397816
            purpose          tot_cur_bal     total_bc_limit      tot_hi_cred_lim         avg_cur_bal         annual_inc
           73.8016848         72.1445531         71.2017339         68.8533554         64.0990039         54.4407981
          totAnnInc         installment             bc_util          catgAnnInc      catgTotAnnInc     mo_sin_rcnt_tl
           53.0114906         52.1618634         45.4046357         43.1125203         33.3555508         30.0688076
  percent_bc_gt_75           open_acc mo_sin_old_il_acct total_il_high_credit_limit      num_op_rev_tl           num_sats
           29.6198645         28.8131666         26.9439235         25.9508680         24.8257371         24.5049183
      home_ownership           mort_acc mo_sin_old_rev_tl_op      num_actv_rev_tl         num_il_tl mths_since_last_delinq
           20.6931492         20.5894246         17.1834883         16.3709402         15.1750233         13.8186595
          revol_bal          num_bc_sats      pct_tl_nvr_dlq num_rev_tl_bal_gt_0 verification_status mths_since_recent_inq
           13.8179036         12.5995407          9.1750814          7.2403074          5.2708094          4.4943527
num_accts_ever_120_pd         delinq_2yrs            pub_rec           open_acc_6m      num_tl_90g_dpd_24m           tax_liens
            4.4497938          1.2440090          1.1389083          0.9679345          0.4340352          0.3534654
```

The above figure also reveals that the three main factors used to determine whether a loan would be charged off or not are the interest rate, grade, and subgrade.

Because they guarantee the best performance when compared to single trees, random forest trees are frequently used by industries.

7. The purpose of the model is to help make investment decisions on loans. How will you evaluate the models on this business objective? Consider a simplified scenario - for example, that you have $100 to invest in each loan, based on the model's prediction. So, you will invest in all loans that are predicted to be 'Fully Paid'. Key questions here are: how much, on average, can you expect to earn after 3 years from a loan that is paid off, and what is your potential loss from a loan that has to be charged off ?

One can consider the average interest rate on loans for expected profit – is this a good estimate of your profit from a loan? For example, suppose the average int_rate in the data is 11.2%; so after 3 years, the $100 will be worth (100 + 3*11.2) = 133.6, i.e a profit of $33.6. Now, is 11.2% a reasonable value to expect – what is the return you calculate from the data? Explain what value of profit you use.

For a loan that is charged off, will the loss be the entire invested amount of $100? The data shows that such loans do show some partial returned amount. Looking at the returned amount for charged off loans, what proportion of invested amount can you expect to recover? Is this overly optimistic? Explain which value of loss you use.

You should also consider the alternate option of investing in, say in bank CDs (certificate of deposit); let's assume that this provides an interest rate of 2%. Then, if you invest $100, you will receive $106 after 3 years (not considering reinvestments, etc), for a profit of $6. Considering a confusion matrix, we can then have profit/loss amounts with each cell, as follows:

		Predicted	
		FullyPaid	ChargedOff
Actual	FullyPaid	*profitValue*	$6
	ChargedOff	*lossValue*	$6

(a) Compare the performance of your models from Questions 5, 6 above based on this. Note that the confusion matrix depends on the classification threshold/cutoff you use. Which model do you think will be best, and why.

The best profit threshold was chosen using a combination of five different cut-offs, as shown in the tables below.

C Threshold	Accuracy	TPR	FPR	F-Score	Profit (using 24)	Loss (using -35)	Overall Profit
0.1	0.4122678	0.1745781	0.6745128	0.1745781	223456	-669258	892714
0.2	0.7845612	0.2578451	0.1345698	0.2578451	588451	-137548	725999
0.25	0.8245126	0.2748514	0.0678451	0.2748514	636451	-67345	703796
0.3	0.8478511	0.2878451	0.0415826	0.2878451	654781	-41785	696566
0.4	0.8545781	0.2845127	0.0214578	0.2845127	667845	-21678	689523

Performance of Boosted tree Model

C Threshold	Accuracy	TPR	FPR	F-Score	Profit (using 24)	Loss (using -35)	Overall Profit
0.9	0.3711	0.172456	0.717264	0.172456	191745	-709457	901202
0.85	0.7347	0.245781	0.217845	0.245781	531784	-213786	745570
0.8	0.8244	0.306871	0.064571	0.306871	631784	-67694	699478
0.7	0.8477	0.339662	0.024578	0.339662	663478	-21813	685291
0.6	0.8638	0.384572	0.001874	0.384572	676896	-1745	678641

Performance of Random Forest Model

The graph demonstrates that Random Forest outperforms Boosted tree by a wide margin.

Due to enhanced AUC and model correctness, Random Forest's overall profit was higher.

As a result, Random Forest will be our top pick for a predictor.

```
> table(pred=predict(DTree3,lcdfTrain, type="class"), true=lcdfTrain$loan_status)
            true
pred         Charged Off Fully Paid
  Charged Off        3627        1280
  Fully Paid         6000       59093
>
```

(b) Another approach is to directly consider how the model will be used – you can order the loans in descending order of prob(fully-paid). Then, you can consider starting with the loans which are most likely to be fully-paid and go down this list till the point where overall profits begin to decline (as discussed in class). Conduct an analysis to determine what threshold/cutoff value of prob(fully-paid) you will use and what is the total profit from different models – decision tree, random forest, boosted trees. Also compare the total profits from using a model to that from investing in the safe CDs. Explain your analyses and calculations. Which model do you find to be best and why. And how does this compare with what you found to be best in part (a) above.

We condensed the data using probability and discovered cumProfit for more investigation.

	scoreTstRF	status	profit	cumProfit
8130	0.9874936	Fully Paid	24	24
29496	0.9873292	Fully Paid	24	48
3446	0.9869573	Fully Paid	24	72
2158	0.9868305	Fully Paid	24	96
25898	0.9867692	Fully Paid	24	120
25969	0.9867161	Fully Paid	24	144
8005	0.9866858	Fully Paid	24	168
19865	0.9866195	Fully Paid	24	192
15676	0.9860481	Fully Paid	24	216
22467	0.9858969	Fully Paid	24	240
15045	0.9857554	Fully Paid	24	264
29125	0.9857511	Fully Paid	24	288
26953	0.9855385	Fully Paid	24	312
15601	0.9855213	Charged Off	-35	277
25500	0.9853611	Fully Paid	24	301
11853	0.9852914	Fully Paid	24	325
14691	0.9852488	Fully Paid	24	349
13554	0.9851170	Fully Paid	24	373
26572	0.9850915	Fully Paid	24	397
28126	0.9850808	Fully Paid	24	421
26614	0.9849045	Fully Paid	24	445
2078	0.9848859	Fully Paid	24	469
5464	0.9848594	Fully Paid	24	493
7906	0.9847969	Fully Paid	24	517
13192	0.9847823	Fully Paid	24	541

Showing 1 to 25 of 30,000 entries, 4 total columns

The following outcome is obtained when we order the results by cumulative profit in descending order.

	scoreTstRF	status	profit	cumProfit
9076	0.5125320	Charged Off	-35	478513
7886	0.5392013	Fully Paid	24	478548
15624	0.5514924	Fully Paid	24	478524
5453	0.5515191	Charged Off	-35	478500
12992	0.5535679	Charged Off	-35	478535
22636	0.5562248	Charged Off	-35	478570
1844	0.5570991	Charged Off	-35	478605
893	0.5621563	Fully Paid	24	478640
2087	0.5636267	Charged Off	-35	478616
14680	0.5656936	Fully Paid	24	478651
21015	0.5705353	Fully Paid	24	478627
17504	0.5744868	Charged Off	-35	478603
3286	0.5755558	Fully Paid	24	478638
16996	0.5777269	Fully Paid	24	478614
931	0.5780071	Charged Off	-35	478590
14606	0.5782837	Fully Paid	24	478625
5510	0.5802304	Fully Paid	24	478601
16318	0.5805387	Charged Off	-35	478577
23319	0.5836702	Charged Off	-35	478612
19215	0.5843402	Charged Off	-35	478647
10274	0.5848328	Fully Paid	24	478682
9590	0.5862362	Fully Paid	24	478658
21305	0.5878570	Charged Off	-35	478634
19681	0.5880913	Charged Off	-35	478669
2482	0.5891515	Fully Paid	24	478704

Showing 1 to 25 of 30,000 entries, 4 total columns

As we can see, 0.5125320 represents the highest likelihood.

It means we can estimate the likelihood score for any future value using this model, and if the probability score is higher than this value, we can invest in them for a higher return.

Let's have a look at it and analyze it.

```
> view(prPerfRF)
> prPerfRF <- cbind(prPerfRF, int_rate=lcdfTest$int_rate)
> prPerfRF <- cbind(prPerfRF, grade=lcdfTest$grade)
> prPerfRF <- cbind(prPerfRF, sub_grade=lcdfTest$sub_grade)
> prPerfRF <- cbind(prPerfRF, annRet=lcdfTest$annRet)
> prPerfRF <- cbind(prPerfRF, actualReturn=lcdfTest$actualReturn)
> prPerfRF <- cbind(prPerfRF, actualTerm=lcdfTest$actualTerm)
> invest_prPerfRF<- prPerfRF[prPerfRF$scoreTstRF>0.5925740,]
> #status wise Avg return, avg int and avg term
> invest_prPerfRF<- prPerfRF %>% group_by(status) %>% summarise(avgInt=mean(int_rate),
+                                                               avgRet=mean(actualReturn),
+                                                               avgTerm=mean(actualTerm))
```

We've added a few new columns to the data frame in the previous section. in the same way as sub grade, int rate, grade, etc.
The average interest rate was ascertained after data with a probability of less than 0.5925740 were eliminated.

```
> invest_prPerfRF
       status   avgInt   avgRet  avgTerm
1 Charged Off 12.02033 5.397854 2.262099
2  Fully Paid 11.98297 5.276888 2.241185
```

Here, we've estimated the typical interest rate. Due to the significant risk associated with a higher interest rate, it does not, however, give a clear image.
Calculate the average return for the invested amount.

```
       status   avgInt   avgRet  avgTerm Deposite_Int investment deposite_return Lading_Club_return
1 Charged Off 12.02033 5.397854 2.262099            2       1000        1045.814             1292.75
2  Fully Paid 11.98297 5.276888 2.241185            2       1000        1045.381             1288.72
```

• Let's assume, we want to invest $1000. Now we have two options

1) Bank Deposit (where we have generally 2% of interest)
2) Lending Club (where we have around 12.06% interest)

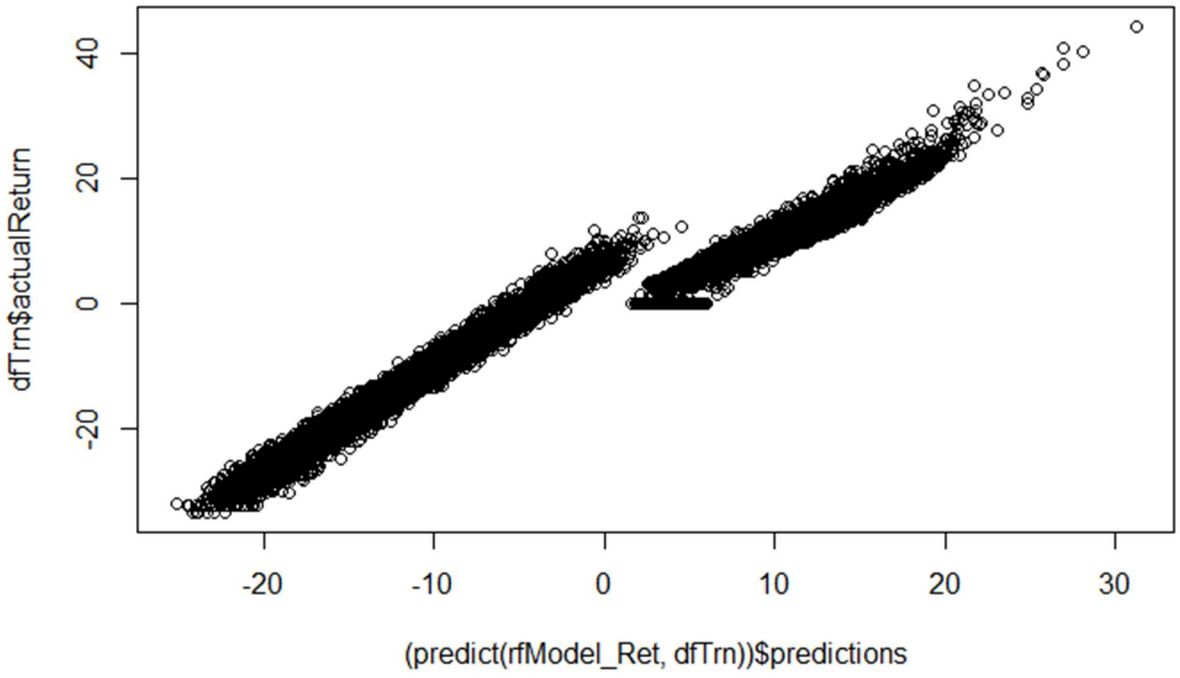
Approximately $1000*(1+(2/100))**2.262172 = 1045.815 will be returned in the first scenario. In the second scenario, though, we can get about $1000*(1+(12.06/100))**2.262171 = 1294.145, which is higher than in the first. Therefore, we ought to use Lending Club to make investments.

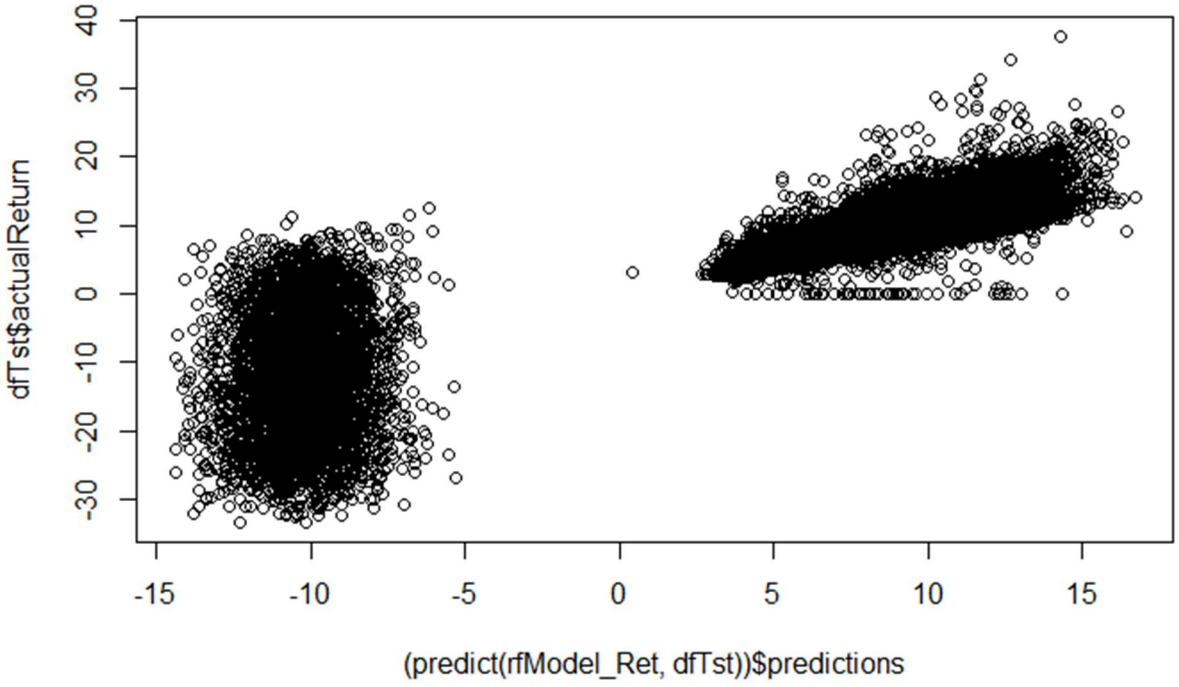
Part B

Part B: predictive models for loans with high returns

8. Develop models to identify loans which provide the best returns. Explain how you define returns? Does it include Lending Club's service costs? Develop glm, rf, gbm (xgb) models for this. Show how you systematically experiment with different parameters to find the best models. Compare model performance.

We have utilized two models to determine the best returns: the Random Forest model and the glm model.
On the training and test datasets, the Random Forest model makes the following predictions.





As seen in the graphic above, our predicted value for the training dataset is generally very close to the actual value. But we didn't get the same result when we tested the dataset.

For the glm model we have received, below value for RMSE.

```
>
> sqrt(mean( (rfPredRet_trn$predictions - dfTrn$actualReturn)^2))
[1] 1.725365
```

9. Considering results from the best model for predicting loan-status and that for predicting loan returns, how would you select loans for investment? There can be multiple approaches for combining information from the two models to make investment decisions (as discussed in class)– describe your approach, and show performance. How does performance here compare with use of single (i.e for predicting loan-status, or loan returns) models?

The GLM model was adopted in this case since it performed the best of all the models. Additionally, two distinct methods were employed to integrate the two models for improved performance.

1) Generated M1 model which predicts loan_status using glm model. Further we divided it in ten tiles and orders it in descending order of average predicted return

tile <int>	count <int>	avgpredRet <dbl>	numDefaults <int>	avgActRet <dbl>	minRet <dbl>	maxRet <dbl>	avgTer <dbl>	totA <int>	totB <int>
1	7000	13.5034725	0	14.720973	9.711951	44.359486	1.337108	0	56
2	7000	10.6766789	0	11.037168	7.605511	16.095350	1.800714	2	798
3	7000	9.3372748	0	9.500885	5.437878	14.031197	1.971454	4	2117
4	7000	8.3091446	0	8.220054	5.036899	11.772826	2.251938	17	3135
5	7000	7.4187738	0	7.257305	2.098622	11.511173	2.369448	174	4619
6	7000	6.4935624	0	6.416298	1.337243	9.946105	2.261841	1172	5706
7	7000	5.5042880	0	5.473203	0.000000	7.656046	2.224546	3330	3647
8	7000	4.5897376	1	4.423897	0.000000	12.281972	2.473447	6005	959
9	7000	0.6419761	2662	2.018250	-9.081518	13.676889	2.807350	4532	790
10	7000	-13.8030176	7000	-16.369475	-33.333333	-1.764495	3.000000	568	1839

2) Similarly, the generated M2 model for predicting actual returns using the glm model. Here also we divided it in ten tiles and orders it in descending order of average predicted return

tile <int>	count <int>	avgpredRet <dbl>	numDefaults <int>	avgActRet <dbl>	minRet <dbl>	maxRet <dbl>	avgTer <dbl>	totA <int>	totB <int>
1	7000	8.099676	1126	8.374626	-32.17619	44.35949	2.184787	35	900
2	7000	6.824745	1066	6.894198	-32.21036	40.24976	2.191229	122	2121
3	7000	6.213847	1032	6.175410	-33.33333	30.14516	2.197408	374	2480
4	7000	5.740574	1031	5.671683	-33.33333	40.81542	2.203859	707	2555
5	7000	5.337736	1020	5.242238	-33.33333	36.86044	2.219856	1153	2626
6	7000	4.960584	947	4.815993	-32.25500	30.67083	2.259159	1662	2639
7	7000	4.589998	963	4.279458	-33.33333	31.32790	2.275064	2127	2567
8	7000	4.206334	841	4.192297	-32.24854	24.22094	2.280170	2648	2635
9	7000	3.756531	814	3.770525	-32.21837	29.24131	2.316644	3177	2597
10	7000	2.968531	823	3.282130	-32.29490	19.60568	2.369669	3799	2546

3) We have selected 1st decile for M2 and ranked by M1 Score

tile2 <int>	count <int>	avgPredRet <dbl>	numDefaults <int>	avgActRet <dbl>	minRet <dbl>	maxRet <dbl>	avgTer <dbl>	totA <int>	totB <int>
1	150	2.435398	13	8.680132	-17.232185	19.295588	2.235834	0	44
2	150	2.366548	8	7.654928	-28.562014	14.315785	2.215378	2	98
3	150	2.476159	9	7.557230	-24.333867	20.649859	2.094579	9	106
4	150	2.491644	13	5.823086	-23.513083	13.770934	2.271791	21	111
5	150	2.583378	7	6.204892	-28.902000	14.805598	2.310495	47	86
6	150	2.535867	5	5.986755	-13.824656	12.672822	2.211211	54	89
7	150	2.579829	11	4.592773	-26.032100	13.341859	2.286913	78	66
8	150	2.777884	7	5.541646	-16.096857	14.246465	2.074967	90	56
9	150	2.517034	9	4.963058	-15.442963	14.477338	2.089742	106	42
10	150	2.741010	6	4.783685	-23.946119	15.035093	2.142487	109	38

1-10 of 20 rows | 1-10 of 14 columns Previous 1 2 Next

Here we can see the average return value has increased as compared to previous ones.

4) Then we multiplied predRetm2 with poScore to find expected return

tile2 <int>	count <int>	avgPredRet <dbl>	numDefaults <int>	avgActRet <dbl>	minRet <dbl>	maxRet <dbl>	avgTer <dbl>	totA <int>	totB <int>
1	150	2.178137	11	8.604915	-17.232185	19.29559	2.205343	1	56
2	150	2.161024	6	7.660126	-22.446044	14.31578	2.146557	13	95
3	150	2.207198	12	6.105972	-17.442267	14.22847	2.307059	22	104
4	150	2.246636	9	6.025081	-13.824656	15.46104	2.239083	55	77
5	150	2.177782	10	4.865272	-28.562014	19.39360	2.238293	75	70
6	150	2.319044	5	5.904303	-26.032100	20.64986	2.081711	80	59
7	150	2.338830	5	5.628530	-23.946119	12.90678	2.122523	95	44
8	150	2.426418	9	4.517452	-24.333867	14.80560	2.266626	98	42
9	150	2.478038	9	4.377163	-28.902000	16.37103	2.222090	106	35
10	150	2.630913	5	4.659662	-25.275833	12.81321	2.260274	110	38

1-10 of 20 rows | 1-10 of 14 columns Previous 1 2 Next

10. As seen in data summaries and your work in the first assignment, higher grade loans are less likely to default, but also carry lower interest rates; many lower grad loans are fully paid, and these can yield higher returns. Considering this, one approach to making investment decisions may be to focus on lower grade loans (C and below), and try to identify those which are likely to be paid off. Develop models from the data on lower grade loans, and check if this can provide an effective investment approach. Compare performance of models from different methods (glm, gbm, rf). Can this provide a useful approach for investment? Compare performance with that in Q9 above?

For this one, we ran GLM after removing grade A and B from our training dataset. In this case, the higher-grade loans have a lower interest rate than the lower-grade loans, which is greater. Additionally, the charged-off loans are higher for lower-grade loans. To put it plainly, the riskier yet higher-interest loans are of worse credit quality. While the loans with a higher grade are safer, their interest rates are lower.

```
       status   avgInt   avgRet  avgTerm Deposite_Int investment deposite_return Lading_Club_return
1 Charged Off 11.99945 5.091451 2.261680            2       1000        1045.805           1292.143
2  Fully Paid 11.99695 5.287073 2.249054            2       1000        1045.544           1290.231
> |
```